Improving Requirements-Test Alignment by Prescribing Practices that Mitigate Communication Gaps

Empirical Software Engineering (2019) 24:2364–2409
https://doi.org/10.1007/s10664-019-09698-6

Improving requirements-test alignment by prescribing practices that mitigate communication gaps

Elizabeth Bjarnason[1] · Helen Sharp[2] · Björn Regnell[1]

Published online: 29 March 2019

Abstract

The communication of requirements within software development is vital for project success. Requirements engineering and testing are two processes that when aligned can enable the discovery of issues and misunderstandings earlier, rather than later, and avoid costly and time-consuming rework and delays. There are a number of practices that support requirements-test alignment. However, each organisation and project is different and there is no one-fits-all set of practices. The software process improvement method called Gap Finder is designed to increase requirements-test alignment. The method contains two parts: an assessment part and a prescriptive part. It detects potential communication gaps between people and between artefacts (the assessment part), and identifies practices for mitigating these gaps (the prescriptive part). This paper presents the design and formative evaluation of the prescriptive part; an evaluation of the assessment part was published previously. The Gap Finder method was constructed using a design science research approach and is built on the Theory of Distances for Software Engineering, which in turn is grounded in empirical evidence from five case companies. The formative evaluation was performed through a case study in which Gap Finder was applied to an on-going development project. A qualitative and mixed-method approach was taken in the evaluation, including ethnographically-informed observations. The results show that Gap Finder can detect relevant communication gaps and seven of the nine prescribed practices were deemed practically relevant for mitigating these gaps. The project team found the method to be useful and supported joint reflection and improvement of their requirements communication. Our findings demonstrate that an empirically-based theory can be used to improve software development practices and provide a foundation for further research on factors that affect requirements communication.

Keywords Empirical software engineering · Software process improvement · Communication · Requirements engineering · Testing · Software engineering theory

Communicated by: Daniel Berry

✉ Elizabeth Bjarnason
Elizabeth.bjarnason@cs.lth.se

Extended author information available on the last page of the article

1 Introduction

Requirements engineering (RE) and testing are two important software engineering processes. They support project success when aligned towards common goals and with well-functioning communication channels between teams and individuals (Bjarnason et al. 2013, Damian et al. 2005, Kukkanen et al. 2009, Martin and Melnik 2008, Post et al. 2009, Sabaliauskaite et al. 2010, Unterkalmsteiner et al. 2014, Uusitalo et al. 2008). In addition, when software artefacts provide the main route for requirements communication, the structure and quality of these artefacts affect the alignment of requirements engineering and testing within a project (Bjarnason et al. 2013).

Software testing requires a clear understanding of the expected behaviour in order to validate that we are 'building the right product' (Boehm 1981) and to verify that we are 'building the product right' (Boehm 1981). This understanding can be provided by RE activities and a clear communication of the requirements (Damian et al. 2005, Damian and Chisan 2006, Bjarnason et al. 2011). However, when the requirements are unclear and ambiguous this can lead to an increased frequency of test failures (Ferguson and Lami 2006). Furthermore, weak alignment of the RE activities and roles with those of software testing may lead to serious implications both for development projects and for the resulting software products. Examples of this include increased development lead time, delayed deliveries, and problems with software functionality and quality (Damian et al. 2005, Damian and Chisan 2006, Uusitalo et al. 2008, Bjarnason et al. 2013).

Defining a generic process that will support the necessary alignment between RE and testing is non-trivial. Apart from the fact that each organisation and project is different and has different targets, how a process is applied depends on how individual engineers function together, i.e. how they communicate. While there is a plethora of methods, frameworks and practices for improving software processes, including CMMI, SPICE etc., methods and techniques for assessing and improving communication within software development are scarce.

We propose a method called Gap Finder for assessing and identifying suitable improvements to the alignment of requirements and testing within a software development project. Our previous empirical research of requirements and test alignment (Bjarnason et al. 2013, 2016) provided the basis on which we designed the Gap Finder method. The method provides a structured and theory-based approach to assessing and mitigating communication gaps. Our method measures *distances* between people and between artefacts, e.g. geographical, cognitive and semantic distances, pinpoints specific distances, or gaps that may negatively affect the requirements communication, and supports identifying relevant practices for mitigating these gaps. For example, the practice of cross-artefact reviews of requirements and test specifications can bridge cognitive distances between the involved roles due to differences in knowledge and insight, and reduce semantic distances between the reviewed specifications, due to differences in meaning.

The empirically grounded theory of distances in software engineering (Bjarnason et al. 2016) was used to design the Gap Finder method by applying a design-science approach (Hevner et al. 2004). In line with this approach, we have evaluated Gap Finder through a case study in which the method was applied to an ongoing IT development project. In this paper, we present the Gap Finder method and report on its formative evaluation. In particular, this paper reports on the following two (previously unpublished) research questions. For the problem of aligning requirements and testing activities:

> RQ1 How relevant do practitioners/participants find the practices identified by the Gap Finder?
> RQ2 How can the Gap Finder be improved to be more useful to an organisation?

The Gap Finder has two main parts: an assessment part and a prescriptive part. The main and novel contribution of this paper is the overall design of the Gap Finder and an evaluation of the prescriptive part of the method (primarily described in Sections 4.1.3, 6.1.2, 6.2.4, 6.2.6, 6.2.7, 7.2, 7.3, 7.4 and Section 8), see Fig. 4. The evaluation of the assessment part and related measurements (see Fig. 1) are published in (Bjarnason and Sharp 2015), and summarised in this paper (primarily in Sections 4.1.2, 4.2, 4.3, and 7.1) to provide a comprehensive description of the Gap Finder method.

The rest of this paper is structured as follows: Section 2 outlines the theory underpinning the Gap Finder, while Section 3 describes related work. In Section 4, the Gap Finder is presented. The case in which the method was evaluated is presented in Section 5, while our research approach including our evaluation method is described in Section 6. The results of the evaluation are reported in Section 7, and discussed in Section 8, in which we also answer the research questions. Finally, we conclude by summarising and describing future work in Section 9.

2 The Theory of Distances

The Gap Finder method supports the application of the previously published Theory of Distances in software engineering that was inductively generated from a systematic literature study and grounded in empirical data from five industrial case companies (Bjarnason et al. 2016). According to this theory, distances between actors, artefacts and activities affect the amount of effort required to coordinate software development. For example, when there is a cognitive distance (or difference) within the team regarding the amount of knowledge of the system under development, additional effort is required to ensure that the requirements are uniformly understood within the project team. Furthermore, software development practices (intended to support development) can mitigate distances, and thus affect the alignment of development efforts. For example, cross-role collaboration in which roles from different disciplines work together on an activity can mitigate cognitive distance by increasing the amount of direct communication.

The theory of distances defines a set of distances, a set of practices, and a model for how practices affect distances. This model is called the Gap Model. The theory is defined to be of interest for software development organisations and projects in general, while the scope of its validity is limited to the case characteristics of the empirical evidence on which the theory is

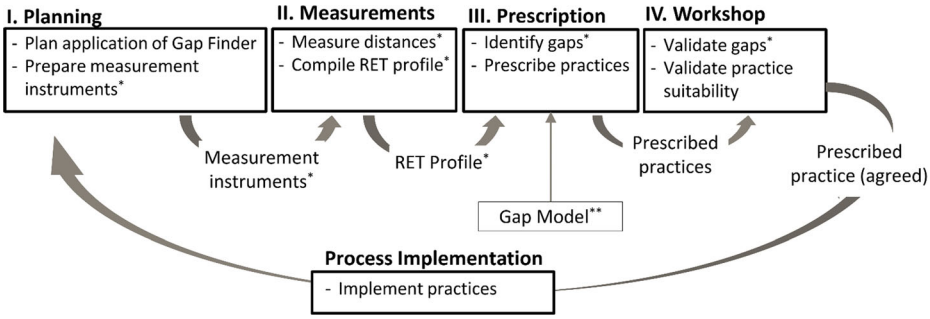

Fig. 1 An overview of the Gap Finder method, its four steps and main output namely measurement instruments, a RET profile and a set of prescribed practices. The parts marked * are also described in Bjarnason and Sharp 2015, in which iRE profile corresponds to RET profile. The Gap Model (marked **, see the Appendix) is part of the Theory of Distances (Bjarnason 2016) and describes how software engineering practices affect distances

based, namely RE-Test alignment for small to medium-sized software development organisations.

The Gap Finder method was designed as a SPI (software process improvement) method for assessing distances and improving the alignment between RE and testing by utilising the knowledge contained in the Gap Model concerning how each practice affects distances. Each of the underpinning building blocks used in the design of the Gap Finder method are described below.

2.1 Types of Distance

A distance is 'a difference in position or level between entities', i.e. actors, artefacts or activities within software development (Bjarnason et al. 2016). A distance requires effort to traverse in order to perform a software development task and can negatively affect the communication of requirements (Bjarnason and Sharp 2015). The theory of distances defines eight types of distance between people, between artefacts and between activities. The distances between people are:

- **D1 Geographical.** The physical distance between the positions of actor's workplaces. For example, a physical distance between a product owner and the testers often has a negative effect on the frequency and ease of communication of requirements. A geographical distance can cause delays and misunderstandings in the communication with distant team members.
- **D2 Organisational**. The distance between actors' placement within an organisational structure, e.g. level within a hierarchy of units and departments. For example, when stakeholders and project members are from different parts of an organisation they may have different objectives and priorities. Organisational distances can cause difficulties and delays in decision making, e.g. concerning conflicting views on which requirements to support.
- **D3 Psychological.** The subjective level of effort perceived to be required by one actor to communicate with another actor. For example, a tester may be reluctant to ask for clarification of requirements if they believe it takes a lot of effort to communicate with the relevant person. Psychological distances can cause conflicts and difficulties in agreeing, e.g. when discussing requirements details.
- **D4 Cognitive.** The difference in levels of cognition between actors, i.e. knowledge, competence and understanding. For example, differences in domain knowledge between a product owner and the development team can lead to differences in understanding of a requirements change. Cognitive distances can also cause both misunderstandings and missed communication.

The artefact-related distances are:

- **D5 Adherence.** The level of similarity between the contents of an artefact and the actual situation. For example, there may be a difference between the produced software and the specified requirements. Adherence distances indicate conflicting understandings of requirements.
- **D6 Semantic.** The level of similarity in meaning between two related artefacts. For example, there is a semantic distance between a requirements specification and test cases when there is a difference in meaning between these two artefacts, which is the case when there is not full test coverage of the requirements. A semantic distance can also indicate conflicting understandings of requirements and failure to keep the artefact updated.

- **D7 Navigational**. The difference in position of related parts of different artefacts required to navigate from one to the other. For example, activities like impact analysis, test coverage etc. require more effort to perform correctly for test cases that are not linked to requirements through traces or similar document structures, i.e. when there is a long navigational distance.

Finally, the activity-related distance is:

- **D8 Temporal.** The difference in time when related activities are performed, e.g. between producing and consuming information. For example, a short time such as a few weeks between defining, verifying and validating a user story and its acceptance criteria, i.e. requirements enables a product owner to catch and adjust requirements misunderstandings early on.

2.2 Practices for RE and Test Alignment

The theory of distances includes eight categories of practices (abstracted practices AP1-8) covering more than 40 industrial practices for improved RE-Test alignment. The eight main categories of practices are

- **AP1 Cross-role collaboration.** The involvement of roles from different disciplines in performing a software engineering activity. For example, involving testers in reviewing requirements.
- **AP2 Separate testers.** The separation of the testers from the roles implementing the software they are to verify to ensure that the verification activities are performed independently.
- **AP3 Documentation.** Documentation of requirements, test case and information related to these.
- **AP4 Aligning document structures and tracing.** The practice of aligning document structures and tracing related entities.
- **AP5 Cross-artefact reviews.** The practice of cross-checking related artefacts against each other.
- **AP6 Incremental software engineering.** The practice of performing software engineer activities incremental, i.e. by iteratively working on a smaller scope and during a shorter time frame at a time.
- **AP7 Automated testing.** Automating one or more of the activities of test planning, design, execution and analysis.
- **AP8 Use of alignment metrics.** The use of alignment metrics, e.g. by the project manager, in monitoring and controlling the progress of a project and the quality of the delivered software.

2.3 The Gap Model

The Gap Model (see the Appendix) pulls together the distances and the practices in a matrix that describes how practices affect one or more distances. For example, applying the practice AP1 Cross-role collaboration can mitigate an organisational distance (D2) between the

involved collaborators by connecting roles from different organisations. This practice can also decrease adherence distance (D5) between an agreed and documented set of requirements by identifying inconsistencies at a review and by updating the requirements documentation accordingly. This theoretical model was derived from empirical data of industrial challenges and practices for requirements-test alignment (Bjarnason et al. 2013). The Gap Model including the full set of identified connections between practices and distances is shown in the Appendix.

The current version of the Gap Model is limited to describing the impact of a practice on a distance type as increasing, decreasing or bridging, i.e. mitigating the negative effect of the distance without changing the distance. For example, AP5 Cross-artefact reviews can bridge organisational distance by bringing together roles from different, i.e. organisationally distant, units to share and discuss perspectives thus mitigating the negative effects of this distance without decreasing the distance itself. However, the model does not provide information about the relative weight of impact of the practices on the different distances or how various distances affect each other. For example, how geographical distance affects psychological distance. Further research is required to explore the impact of practice in more detail, and potential relationships and correlations between distances.

3 Related Work

The aim of our research is to improve the alignment of requirements engineering and testing by providing knowledge and methods for improving software processes and practices.

3.1 Aligning Requirements Engineering and Testing (RET)

Even though the alignment of RE and testing is a significant challenge within software development (Bjarnason et al. 2013) there is a limited amount of research into the combined area of requirements-test alignment, rather most research tends to focus on one area, i.e. on RE or on testing (Barmi et al. 2011). Of the research published in the area of requirements-test alignment Barmi et al. (2011) found that most studies were on model-based testing in which test cases are generated from formal models of requirements. Barmi et al. also identified a need for research into traceability and empirical studies into alignment challenges and practices. Only three empirical studies specifically focusing on requierments-test alignment were found. Of these, two originate from one research group (namely Kukkanen et al. 2009 and Uusitalo et al. 2008), and the third one is from our previous requirements-test alignment study (Sabaliauskaite et al. 2010, Bjarnason et al. 2013).

Our study (Bjarnason et al. 2013) identified all of the alignment practices reported by Uusitalo et al. (2008) with the exception of the practice of linking testers to requirements owners and the practice of including internal testing requirements in the project scope. Their study also reports that linking people is equally important to linking artefacts (Uusitalo et al. 2008). Similarly, Kukkanen et al. (2009) report that requirements-test alignment is supported by connecting processes and people, and by applying good practices, and advocate assigning specific roles for requirements management and for test management, that are responsible for coordinating between the two areas (Kukkanen 2009). The importance of connecting people is confirmed by our requirements-test alignment study that concludes that communication is one of the major challenges in achieving good alignment (Bjarnason et al. 2013). In this context, it is interesting to note that Gotel and Finkelstein (1994) express that a particular concern in improving requirements traceability is the need to facilitate informal communication with those

responsible for specifying and detailing requirements, i.e. that interactions plays a key role in enabling requirements-test alignment.

3.2 Software Process Improvement (SPI)

The field of software process improvement (Humphrey 1989, 1997, Basili and Rombach 1988) is rooted in the perspective that 'the software process is the set of tools, methods, and practices we use to produce a software product' (Humphrey 1989, p.3.) The process, including its practices, is then an important instrument in developing and maintaining quality software products in an efficient, reliable and repeatable way. Improving the practices applied in a development project can thus improve the efficiency and effectiveness of the software engineering efforts.

There is a wide range of SPI frameworks, or methods, which in general share the same main steps of first evaluating the current process, and then identifying, implementing and evaluating suitable process improvements. These frameworks may be categorised into two main approaches: inductive and prescriptive (Briand et al. 1995). Inductive (or bottom-up) frameworks, such as QIP (Basili 1985) and Lean Six Sigma (George 2002), take their stance in the organisational situation and context of the organisation when identifying potential improvements. In contrast, prescriptive (or top-down) frameworks, such as CMM/CMMI (Chrissis et al. 2007), People CMM (Curtis et al. 2002) and SPICE, i.e. ISO/IEC 15504 (ISO/IEC 2004-2011), mainly base their improvement suggestions on a wide set of best practices.

Retrospective reflection is another example of an inductive approach to SPI. At retrospective meetings, project members identify problems and potential improvements to their practices by considering and analysing past events and experiences (Collier et al. 1996, Derby and Larsen 2006, Drury et al. 2011). By providing a time line of project events in these meetings, a project team can gain a joint understanding of the overall project and connections between roles and activities throughout the life cycle (Bjarnason et al. 2014), which is beneficial for improving project collaboration and communication.

3.3 SPI Methods for Requirements-Test Alignment

There are SPI methods that address RE (Lavallée and Robillard 2012) and testing (Afzal et al. 2016) separately, but very few explicitly consider the alignment between these two areas. Gap Finder is one exception to this, and REST-bench another. REST-bench was designed by Unterkalmsteiner et al. (2014) to assess and improve requirement-test alignment by modelling the information flow between requirements and testing for a specific project using an artefact map that is elicited individually from requirements and testing roles. This approach enables uncovering inconsistencies and incongruences in viewpoints between requirements and testing roles. These roles then discuss and resolve these misalignments at a common workshop. When applying REST-bench on a one-year project at Ericsson AB, a number of misunderstandings were uncovered and subsequently resolved at the workshop, which also resulted in identifying bottlenecks and sub optimisations in the requirements-test interaction (Unterkalmsteiner et al. 2014).

The current version of CMMI (Chrissis et al. 2007) includes process areas for RE, for validation and verification, and describes intended connections between these process areas and includes alignment practices such as traceability and cross-review of requirements against project plans and other work products, e.g. design and test artefacts, managing requirements changes. These practices have been identified as supporting requirement-test alignment (Bjarnason 2013,

Uusitalo et al. 2008) and applying them as part of a CMMI effort would thus be expected to lead to an improvement. Similarly improving the quality of the requirements through SPI has been found to improve the overall quality of the software product (Kandt 2009; Bjarnason et al. 2013). However, Harter et al. (2012) found that there was a significant decrease in the rate of severe defects at higher CMM levels, but that this effect was not present for projects with a high degree of requirements ambiguity.

We are not aware of any empirical research studies on the impact of CMMI on requirement-test alignment. However, there are some studies that report on the correlation between the RE and the testing processes. Damian and Chisan (2006) found that simultaneously improving these processes can lead to pay-offs in improved test coverage and risk management, and in reduced requirements creep, overscoping and waste, resulting in increased productivity and product quality (Damian 2006).

3.4 Theory-Based SPI Methods

We find few SPI approaches that are based explicitly on theories, as the Gap Finder is. One exception is the team radar instrument initially proposed by Brede Moe et al. (2009). The team radar is based on an empirically-based theory of team-work challenges and aims to improve agile software development projects. This instrument can be used to qualitatively assess factors defined by the theory as influencing team work. Similarly to the Gap Finder these factors are measured and rated through qualitative data collection, and then presented to the development team as a visual radar diagram. Improvements are then identified by jointly reflecting on these underlying factors. The instrument was found useful to practitioners in identifying improvements and the five factors (from the underlying theory) were confirmed as relevant to team work in agile development.

The team radar instrument (Brede Moe et al. 2009) was further improved by Angermo Ringstad et al. (2011) through strengthening the diagnosis step and by applying action planning. The diagnosis phase of the method was expanded to also include observations, in addition to interviews, of the assessed team's daily work. The rating of the underlying factors for team work was based on a structured analysis of all the gathered data, i.e. interview transcripts and field notes from the observations. An action plan to address the issues found was then specified at a meeting in which the ratings were presented to the team who were invited to discuss the presented picture and areas to improve on. The defined actions were based on the underlying theoretical framework of factors affecting team work in agile development. This improved version of the team radar was found to support project teams by illuminating issues not previously discussed within the team. This contributed to providing a view of the situation by highlighting underlying factors and causes, rather than merely pointing out experienced problems.

4 The Gap Finder Method

The Gap Finder enables the assessment of a development project and the identification of relevant practices for improving requirements-test alignment. It consists of two parts: an assessment part based on a set of measurement instruments for assessing distances within a development project; and a prescriptive part that supports identifies improvement practices utilizing the Gap Model (Bjarnason et al. 2013), see Section 2.3. The measurement instrument enables the development of a RET profile of the current level of requirements-test alignment for the project under study. The Gap Model is then used to pinpoint improvement practices that

can bridge or decrease the troublesome distances identified through the RET profile. These practices are identified by comparing the distances found in the obtained RET profile with the Gap Model, extracting practices known to mitigate the identified distances and constructing a suitable set of improvement practices for the development project. The outcomes, i.e. the RET profile and the identified improvement practices, are presented to the assessed project team at a workshop with the dual purpose of validating the output of the method and agreeing which improvement practices to implement.

The main steps for applying Gap Finder are described in Section 4.1. The measurement instruments are presented in Section 4.2 and the resulting RET profile is described in Section 4.3; these are also described in (Bjarnason and Sharp 2015). Guidelines for applying the Gap Finder are available on-line (Bjarnason 2013).

4.1 The Four Main Steps

Gap Finder involves four main steps: I) planning for assessment, II) measurement of distances within project, III) prescription of improvement practices and IV) workshop to validate outcome. The agreed practices are then implemented and the project is re-assessed by iterating the steps. An overview of these steps is shown in Fig. 1.

4.1.1 Step I: Planning

The scope, extent and timeframe of the assessment are planned in agreement with the host organisation in which the Gap Finder is to be applied. In addition, the measurement instruments of the Gap Finder need to be prepared for the assessment. Both of these activities require insight into the processes and practices of the organisation. The method could be applied by someone with this insight, e.g. a process engineer but if performed, e.g. by a researcher, initial investigations are needed to obtain this knowledge. In particular, knowledge of roles and artefacts involved in the requirements and testing processes is needed.

The planning includes identifying which part of the organization to assess and which roles and individuals to include. For example, one team or sub-set of roles could be selected for assessment. The people and relevant artefacts to include in the assessment can then be identified and agreed.

The measurement instruments may need adapting to the processes and terminology specific to the assessed organization. The instruments thus need to be reviewed and possibly revised to refer to case-specific terminology, e.g. names of specific roles, artefacts etc. This will reduce misunderstandings and support a more consistent measuring and understanding of the assessed factors.

The output of the planning step is the measurement instruments of the Gap Finder adapted to the specific case, and an agreed plan for the assessment.

4.1.2 Step II: Distance Measurements

Gap Finder's measurement instruments consist of three questionnaires: profile, communication and artefact questionnaires. The profile and communication questionnaires contain questions concerning the project members, while the artefact questionnaire investigates distances between the requirements and test specification. The instruments are further described in Section 4.2

The questionnaires are administered to the roles involved in the requirements and testing activities. The first time Gap Finder is applied to an organisation, it is recommended to use interviews for the surveys. This will allow the participant to ask for clarifications, which can enable a uniform understanding of the questions and of the scales used to answer them. In addition, the interviewer can ask follow-up questions and thereby obtain a richer picture of potential issues and reasons for them. This is particularly important when the interviewer is not intimately acquainted with the project.

The results of the distance measurements are collated into a RET profile for the assessed project (see Section 4.3). This profile is then used in the prescriptive step and presented at the workshop.

4.1.3 Step III: Prescription

In this step, the RET profile is analysed to identify gaps. When long distances and potentially troublesome gaps are diagnosed, improvements are prescribed based on the underlying theory. In particular, the Gap Model (see Table 5 in the Appendix) is consulted to identify software development practices that can mitigate the identified gaps. This set of practices is then further refined and a final set is identified, adapted to the assessed organisation.

The analysis is supported by any additional knowledge about the case, e.g. contextual factors such as project size, development model, specific practices applied. For example, if a large organisational distance is seen in the RET profile, the Gap Model suggests 3 categories of abstracted practices for mitigating this organisational distance, containing in total 34 individual practices. This large set can be whittled down to a more manageable number by a combination of matching the sets proposed by Gap Model for each identified gap and considering the suitability including cost of each practice for the assessed development organisation. The aim is to identify a small set of practices that can mitigate all the identified gaps and that are a good match for the organisation at hand.

4.1.4 Step IV: Workshop

The RET profile and the set of prescribed improvement practices are presented to the project team at a workshop. For each type of distance, the relevant parts of the RET profile including the gaps are shown and the suggested improvement practices are presented. The project members are encouraged to share their observations of potential issues caused by the identified gaps and if and how the suggested practices may mitigate them. This allows for a validation of both the gaps and the practices identified through applying Gap Finder. Furthermore, it includes the project members in the decisions regarding which improvements to implement thereby increasing the probability of successfully implementing the new practices.

4.1.5 Implement Practices and Iterate

After the agreed practices have been implemented, the situation is re-assessed by iterating the steps. A new assessment plan is made (step I), the distances are re-measured (step II) and another prescription (step III) is derived. In this analysis, the original and the new RET profiles are compared to assess if the previous gaps have been reduced and/or whether the effects of them have been minimised by the implemented practices. Additional or different improvement practices may be uncovered through analysis of the new RET profile. These are then reviewed and discussed with the project team at another workshop (step IV). At this session a decision is

made as to whether or not the SPI effort is completed, and if not the Gap Finder process is iterated again.

4.2 Measurement Instruments

The measurement instruments used for assessing a project contain 18 measurements (see Table 1) that cover eight types of distance. These measurements are applied to artefacts and people involved in the requirements and testing activities. While some distances are straightforward to assess, others are estimated through questionnaires with self-rating questions. For example, geographical distance (D1) is assessed by measuring the physical distance to walk between desks, while psychological distance (D3) is measured through a questionnaire asking each team member to rate the distance between themselves and each other member of the team.

Most of the questions have Likert scales with five options for the respondent to choose between. For example, for psychological distance (D3, M3.1) the respondents were asked to rate how hard it was to communicate with colleague *n* by noting 1-5 for *Not hard* (1), *Some effort required* (2), *Medium effort* (3), *Much effort* (4), *Extremely hard* (5). Similarly, for the knowledge aspects of cognitive distance (M4.1-M4.3) the respondents were asked to grade their own competence using Benner's (1982) five levels of experience, i.e. *Novice* (1), *Advanced beginner* (2), *Competent* (3), *Proficient* (4) and *Expert* (5). The cognitive distance between two people was then measured by calculating the difference between their levels of competence.

The distance types are complex and can contain several aspects. For five of the distance types we have defined one measurement per aspect and, thus, several measurements per distance type. For example, for cognitive distance (D4) five aspects are measured: one aspect of prioritisation of quality aspects for the system, and three aspects of different types of knowledge: domain; technical skill; organisation; and process.

For the artefact questionnaire, the aspects of abstraction (M5.2.3, M6.3) and coverage (M5.1.2, M5.2.2, M6.2) are directional, i.e. the abstraction level of artefact A may be higher or lower than artefact B. For these questions the following scale was used: *Much more*, *Somewhat more*, *The same*, *Somewhat less*, *Much less*, and *Can't say*.

The aspect of priority for cognitive distance (M4.4) was assessed with a question on the relative priority of the quality characteristics specified in ISO/IEC 9126-1. The respondent was asked to distribute 30 resources over the six quality characteristics. The distance (of M4.4) between two people was then assessed by calculating the Cartesian distance between their responses.

The distance for the measured aspects can be calculated in various ways either individually per measurement or combined to a total distance for the whole project. For example, the average value for one aspect of distance between each pair of team members can be considered, or the distance between the minimum and the maximum value. The total for a distance type consisting of multiple aspects can be obtained by calculating the Cartesian distance between the multi-dimensional data points for each aspect and participant.

4.3 The RET Profile

We pose that a project's current level of requirement-test alignment can be assessed by considering its RET profile,[1] i.e. distances between RE and testing roles and activities. The

[1] Previously called iRE profile in Bjarnason et al. 2016

Table 1 Outline of the Gap Finder measurement instruments and the results obtained for the case project

Measurement		Min	Average	Max	Distance	Aspect
Profile Questionnaire						
M1	Physical distance between desks (metres)	1.8	76.7	322	D1	Physical
M2	Length of path in line organisational tree between two people (steps)	0	2.6	7	D2	Home unit in line organisation
Communication Questionnaire						
M3.1	Perceived effort to communicate with another person	0.20	0.35	1.00	D3	Uni-directional
M3.2	Perceived effort to communicate between two people	0.20	0.35	0.60	D3	Bi-directional
Profile Questionnaire						
M4.1	Difference between people's knowledge of system domain	0.00	0.32	0.80	D4	Domain knowledge
M4.2	Differences in competence within technical areas affecting requirements and testing alignment	0.17	0.32	0.50	D4	Technical skill
M4.3	Differences in knowledge of project and organisation including processes	0.06	0.32	0.56	D4	Process and organisation
M4.4	Differences in prioritisation around product	0.03	0.08	0.14	D4	Priorities
Artefact Questionnaire						
M5.1.1	Difference between product actual and agreed product behaviour	0.00	0.17	0.25	D5.1: Delivered vs agreed reqts	Similarity
M5.1.2	Difference in coverage between actual and agreed product behaviour	0.00	0.00	0.00	D5.1: Delivered vs agreed reqts	Coverage
M5.2.1	Difference in meaning between documented vs agreed requirements	0.00	0.00	0.00	D5.2: Agreed vs documented reqts	Similarity
M5.2.2	Degree of coverage between documented vs agreed requirements	0.00	0.00	0.00	D5.2: Agreed vs documented reqts	Coverage
M5.2.3	Difference in abstraction level between documented vs agreed requirements	0.50	0.67	1.00	D5.2: Agreed vs documented reqts	Abstraction
M6.1	Difference in meaning between requirements and testing artefacts	Roughly the same*			D6: Reqts vs test cases	Similarity
M6.2	Degree of coverage between requirements and testing artefacts	Somewhat more*			D6: Reqts vs test cases	Coverage
M6.3	Difference in abstraction level between requirements and testing artefacts	Somewhat more*			D6: Reqts vs test cases	Abstraction
M7.1	Number of clicks to navigate from a requirement to the test cases that verifies it	Not included in case study			D7	Reqt to Test cases
M7.2	Number of clicks to navigate from a test case to the requirement(s) that is verifies	Not included in case study			D7	Test case to Reqts
M8	Length of time between specifying a requirement and defining a test case to verify it	Not included in case study			D8	Reqts – Test case definition

The instruments consist of the measurements (M1-M8) per distance (D1-D8) and the questionnaires in which they appear. All measurement values are normalised within the range of 0–1, except for geographical and organisational distance. * indicates only one data point for measurement

RET profile is produced by collating the measurements for each distance. For example, the cognitive and psychological distances between the roles responsible for requirements and testing are included in the RET profile.

The range and average value for each type of distance can be presented as part of the project's RET profile. For measurements with the same scale, or scales that can be normalised, the various aspects and distances can be visualised together in a radar diagram, see example in Fig. 2. In order to avoid the limitations of this type of visualisation, the ordering of the axes needs to be considered and kept consistent, in particularly when comparing diagrams over time. Furthermore, the relative difference between the different distance types should be considered when analysing these diagrams rather than the total area made up of all data points.The RET profile is used as input to the prescription step (step III) and to the workshop (step IV). When analysing the RET profile individual distances between project members and roles may need to be considered to identify distances that need addressing. Similarly upon re-assessing a project, the two versions of the RET profile can be compared to assess the effect of the implemented practices.

5 Case Description

A development project within The Open University's IT unit provided the case for this study. The Open University is UK's largest academic institution with more than 240,000 students from all over the world. The IT unit is responsible for the day-to-day management of the university's information systems and in-house development of some systems. The studied project is part of a programme developing a system for student administration and curriculum management to meet the new requirements posed by evolving curriculum needs, changed fees and funding regulations, and subsequent changes to internal business processes. An overview of the case is provided in Table 2.

The Scrum development method is applied at team and intra-team levels. Each development team consists of a product owner, a requirements analyst, a tester, a number of developers and a scrum master. In addition, there is a project manager responsible for the project to which the team delivers. The product owner represents the business and is responsible for the scope including signing off on acceptance of project deliveries. The requirements analyst is responsible for eliciting and defining the requirements in close collaboration with the product owner

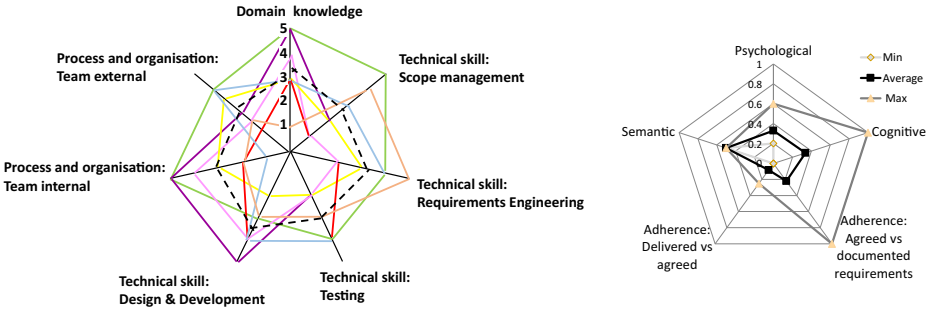

Fig. 2 Examples of radar diagrams used in the Gap Finder workshop to provide a visual overview of the multiple distances measured for the case project. The left-hand diagram shows aspects of cognition; the level for each member (coloured lines) and the average value (dashed line). The right-hand diagram shows the normalised min, max and average values for several distances within the whole project team

Table 2 Characteristics of the studied case at Open University's IT department

Type of case	Academic education provider
# people in software development unit	Approx. 150 for IT development (300 for whole IT unit)
# people in project	Approx. 20
Distributed	No
Domain / system type	IT: Educational programme management including student services
Source of requirements	In-house
Main quality focus	Maintainability
Certification	No
Process model	Scrum
Duration of project	2–3 years
# requirements in project	Approx. 800 user stories
# test cases in project	Approx. 1300 test cases
Product lines	No
Open source	No

and the development team. The scrum master, project manager, developers and testers all take an active part in discussing, and thereby defining, the requirements. Finally, the tester within the team is responsible for verifying that the produced software corresponds to the requirements. The team members of the studied development team are characterised in Table 3.

The project scope is described in definition documents and in agile epics by senior requirements analysts and allocated to one of the four planned system releases. For each release, the requirements analyst for the intended development team details the epics into user stories and acceptance criteria and places these in the team's backlog. Development is performed in 2-week sprints (iterations) and prior to each sprint the user stories in the backlog are prioritised by the product owner and requirements analyst. The user stories with the highest priority are then presented to the development team who estimate them. A set of stories are agreed on for that sprint according to priority and team capacity.

The epics, user stories and acceptance test cases are stored in a central requirements repository with traceability links. The test scripts are stored in another repository and linked to the relevant user story. These test scripts can then be viewed from the requirements repository.

Once the development team has delivered accepted functionality, the system is tested as a whole both from a user perspective and from a system integration perspective. This testing is performed by team-external testers and by representatives from the business unit. Any issues found in this testing is initially analysed by the tester in the development team before further

Table 3 Roles and length of experience for the members of the studied development team

Roles	Length of experience in team (months)	Length of experience with case organisation (years)	Length of experience in current role (years)	Total length of work experience (years)
Product owner	10	16	8	26
Requirements analyst	0	24	24	28
Tester	3	0.25	10	26
4 developers	8, 9, 9, 0	4, 1, 6, 0	6, 5, 22, 10	7, 6, 22, 10
Scrum master	10	17	1	26
Project manager	3	0.25	18	25

decisions and actions are taken to either reject or agree to address the issue. The team tester and the team-external testers are assigned from the same department.

6 Research Method

The Gap Finder method was designed through a design science research approach (Hevner et al. 2004) and formatively evaluated using a case study method (Runeson et al. 2012). The formative evaluation of the Gap Finder method aimed to seek in-depth feedback that could guide further design and improvement of the Gap Finder and thereby ensure that the method is usable and useful (Rogers et al. 2011). Due to this being a formative evaluation, the Gap Finder and the case study were iteratively designed throughout the study. The design science components, i.e. environment, design science research, and knowledge base, are outlined in Figs. 3, and 4 contains an overview of our research method.

We performed this research in three main stages, namely design, evaluation and data analysis using a combination of empirical research methods. Apart from the methods included in the Gap Finder method (i.e. questionnaires and workshop, see Section 4), observations, a survey and interviews were also performed. An ethnographically-informed approach (Robinson et al. 2007) was taken in the observations to ensure that relevant data was collected with the Gap Finder and that it was understood in-line with the team members' perceptions of the situation. Figure 4 provides an overview of our research method and highlights the main novel parts of the research presented in this paper.

The design of the Gap Finder, study design, data collection and analysis were mainly performed by Bjarnason, and reviewed and validated by Sharp. In addition, Sharp provided support in the contact with the case organisation, participated in one initial interview and in the workshop where the outcome of the Gap Finder was presented to the development team, and conducted the post-study interviews. The workshop was held as a focus group session.

6.1 Design

This stage included the design of the Gap Finder and of the case study through which it was evaluated. Both of these activities require insight into the case organization to be used for the evaluation. In particular, knowledge of the roles, artefacts and practices of the live development project for which Gap Finder was to be applied was needed.

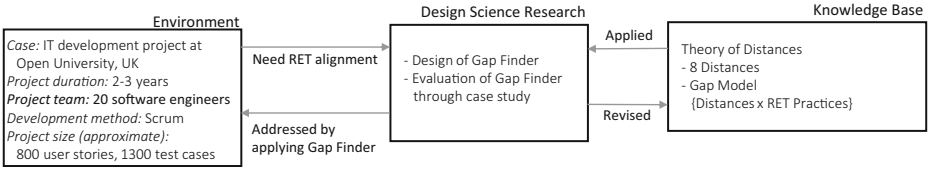

Fig. 3 An overview of how a design science research approach (Hevner 2004) was applied in designing and evaluating the Gap Finder method for a case environment based on the Theory of Distances for Software Engineering (Bjarnason 2016)

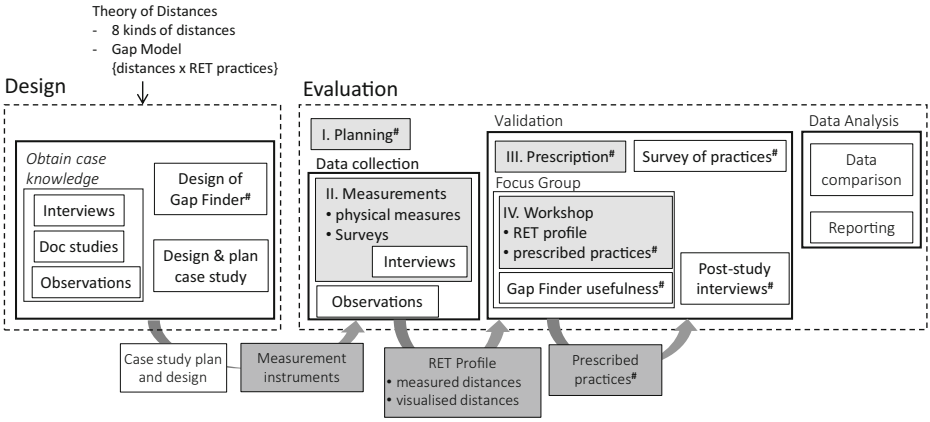

Fig. 4 An overview of the performed design science research including the main intermediate artefacts and data. Parts directly related to applying the Gap Finder method are marked with grey; light grey for activities and dark grey for artefacts and data. # denotes the main novel contributions of this paper relative (Bjarnason and Sharp 2015), namely the overall design of Gap Finder, and evaluation of the prescriptive parts of the method

6.1.1 Obtaining Knowledge of Case

Initial knowledge of the case was obtained through document studies, a semi-structured interview, demonstrations, and observations of the development team prior to applying the Gap Finder. One of the authors had an existing relationship with the studied organisation and some initial documentation and contacts. The organisation provided further project documentation upon request and the first author studied these. These documents included information about project organisation, i.e. roles and names, the applied development processes, and examples of artefacts. In order to obtain further insight into development artefacts, the artefacts used for testing were demonstrated to us, and the first author was granted access to the systems used for development to study additional examples of requirements, test cases, defect and test reports.

We designed an interview instrument that is available on-line (Bjarnason 2013) to obtain further knowledge about the roles, artefacts and activities used for RE and testing. Two managers within the IT development unit agreed to participate in this initial open semi-structured interview. The managers shared their view of current challenges and good practices and supplied a number of pointers to information and people.

Finally, an initial observation of the development team was performed. One researcher was present in the team area for a consecutive period of three days at the end of one sprint, including review and planning meetings for the next sprint. The researcher observed team members, what they were working with, and their rapport with each other. The researcher did not interact or disturb the project members, but merely observed how, with whom and about what they interacted. This allowed the researcher to gain familiarity with the team and with their day-to-day work. This insight enabled detailed design of the measurement instruments and of the research method. Furthermore, this initial observation established a relationship between the development team and the researcher that facilitated further data collection.

6.1.2 Design and Evolution of the Gap Finder

An iterative approach was taken to design the Gap Finder as a method to be applied in practice. The initial version of the Gap Finder was designed using the existing knowledge base, in particular the theory of distances (Bjarnason et al. 2016). Knowledge of the case environment was used to evolve the subsequent version of the method. Further refinements were identified through discussion in the research team and as new insight was gained from applying it to the case. For example, later parts of the method such as constructing the RET profile, prescribing improvements and the workshop were designed as the application of the method to the case progressed.

The Gap Finder method evolved through a combination of generic design based on the theory of distances and designing for the specific case. For example, generic distance measurements were designed for each of the distance types defined by the theory, and then detailed and realised through the aspects and roles believed to be relevant for the case, e.g. product owner, requirements analyst, and development team tester. The aim was to design a practical method for the case environment, while keeping the design generic enough to be applicable also to other software development organisations and projects by aligning it with the underlying theory and related work.

The final step, i.e. the workshop step IV was designed as a focus group (Robson 2002) involving all team members of the assessed project since involvement is a known vital factor in effectively implementing SPI changes (Dybå 2000). The intention of this design decision is to inform team members of factors underlying the targeted issues and to obtain a consensus and commitment to a set of improvement practices.

6.1.3 Case Study Design and Planning

Gap Finder was applied at the sprint iteration level for one development team. This allowed an evaluation of the method covering a full set of development activities including requirements detailing, design, development and testing within a feasible time frame and with a clearly delimited set of requirements. Data was collected through interviews, observations and a final survey to assess the suitability of the prescribed improvements.

An explorative research approach was taken in the design of the case study meaning that the initial study design evolved and was adapted over time as new insights were gained. Similarly, even though the different parts of the case study are described here as separate and sequential activities, an iterative approach was applied throughout the study, so the study design was continuously re-visited. For example, when it became apparent that the applicability of certain practices had not been commented on during the Gap Finder workshop, a survey of the suggested practices was designed to complement that data.

6.2 Evaluation: Data Collection and Validation

We evaluated Gap Finder through a case study in which the Gap Finder method was applied to an ongoing software development project. Evaluation-specific data collection was performed through semi-structured interviews that were held in connection with measuring the distances (Gap Finder: Step II) and by observing the team for the time period during which Gap Finder was applied. In addition, the workshop (Gap Finder: Step IV) was held as a focus group and extended to also validate the method, and a survey was included to assess the prescribed improvements. This additional data allowed the researchers to evaluate the Gap Finder including the measurements and the set of prescribed practices.

6.2.1 Gap Finder Step I: Planning

Before applying the Gap Finder, the assessments needs to be planned and the measurement instruments prepared. The information obtained about the case organisation and project (see Section 5) was utilised for this. In particular, this included knowledge gained about which roles and artefacts were involved in the requirements and test activities, and how the requirements and test cases were managed in the requirements repository. Based on this insight the researchers decided to exclude the measurements for navigational (D7) and temporal distance (D8) from the evaluation study. Since tracing was applied between requirements and test cases the navigational distance would always have resulted in the value 1, which would not show any gaps for this dimension. Temporal distance was excluded from the evaluation due to the practical difficulties in measuring this for the case organisation where face-to-face communication was the main source of requirements information rather than artefacts. The measurement instrument used for the evaluation (with some terms replaced for confidentiality and anonymity reasons) is on-line (Bjarnason 2013).

Since the requirements were defined through team discussions in this agile project, all roles represented in the development team were involved in requirements activities. However, the product owner, requirements analyst, tester and developers were the primary roles involved, while the scrum master and project manager were primarily involved in requirements discussions at a more general level. The measurement instrument was prepared accordingly. Namely, the measurements were adapted to cover all of the primary roles and their corresponding technical skills of scope management, requirements engineering, testing, design and development. In addition, all roles were asked to participate in the profile and communication questionnaires, while the artefact questionnaire was limited to the product owner, requirements analyst and tester.

Similarly, the measurement instrument was designed to cover the specific artefacts and activities used for requirements and testing in the case organisation. In addition, generic terminology was replaced with specific terms used within the organisation, e.g. ‘documented requirements’ was replaced with ‘user stories’.

6.2.2 Gap Finder Step II: Distance Measurements

The distances within the development team and between their requirements and testing artefacts were assessed using the measurement instruments prepared in step I. The majority of the people-related distances were assessed through the profile questionnaire, which covered each role within the team. In addition, psychological distance between each pair of team members was assessed through the communication questionnaire, which was taken by all team members. The artefact-related distances were measured at the end of the second sprint by administering the artefact questionnaire to the product owner, requirements analyst and the tester. These distances were assessed for the specific requirements and test cases related to the functionality delivered in that sprint. For each questionnaire, the targeted respondents were free to choose whether or not to participate.

The measurement questionnaires were administered as semi-structured interviews to ensure uniform understanding of the questions and to enable collecting richer data for evaluation purposes. These interviews took around 15-30 minutes each and were audio recorded and transcribed.

6.2.3 Observations

In parallel to applying the method, additional data relevant to project communication and distances was gathered through ethnographically-informed observations. These entailed seeking to understand the team's work practices apart from the researcher's assumptions about software development (Robinson et al. 2007) and enabled the researchers to gain a rich insight into the day-to-day work practices and interactions of the project members. Collecting this additional data had the dual purpose of enabling triangulation and of providing insight into issues that the team experienced, and strategies applied to mitigate them.

The observations were as unobtrusive as possible and questions were only asked to seek clarification of terminology or actions, and never to participate in team discussions. The set of distances provided a 'protocol' for the observer in taking particular note of interactions taking place in the team area. Extensive field notes were made during the observations including the nature of interactions, status of ongoing work and information shared during meetings, and individual activities.

6.2.4 Gap Finder Step III: Prescription

Gaps were identified by analysing the obtained measurements and identifying practices for mitigating these through the Gap Model. The measurements were then collated into a RET profile for the assessed project team.

The gaps were identified by analysing the measured data from a number of different perspectives. A qualitative approach was taken in analysing the quantitative measurement data. In future, when reference data from other cases becomes available a quantitative aspect could be added to this analysis. However, contextual factors always need to be considered since these may influence if a distance is 'good' or 'bad'. For example, for a case with extensive requirements documentation the distance between artefacts is likely to be more critical than for a case relying heavily on face-to-face communication of requirements.

The analysis of the distances entailed calculating the distance per measured aspect between each team member and the total distance for each type of distance. In addition, for each distance measurement the average and range of obtained values were calculated and analysed. For the measurements using Likert scales, the median values were calculated and compared to the mean values. As they were very close, the means were used in order to present uniform types of value. Since the further analysis of these values was qualitative, we judge that this choice did not affect the following Gap Finder steps. The distance between pairs of people was also calculated to identify potentially large gaps between specific roles and individuals. From this, a set of gaps to mitigate was identified.

Corresponding improvement practices were identified and prescribed as described in Section 4.1.3. An initial set of abstracted practices were pinpointed based on the Gap Model for the identified distance types. Specific practices were identified by considering additional data and insight into the case, thereby customising the theoretically-derived practices for the specific case. For example, the largest geographical distance within the team was due to the product owner being seated in a different building from the rest of the team. The Gap Model states that geographical distance can be decreased through the practice of cross-role collaboration (AP1). This abstracted practice was then refined into the more specific practice of increasing the product owner's physical presence to developers and testers by providing a guest desk for this person. Furthermore, since several practices affect the same distance types

the set of improvement practices was designed to provide a good coverage while avoiding conflicting practices.

6.2.5 Gap Finder Step IV: Workshop & Focus Group

A Gap Finder workshop was held, as a focus group, at the beginning of the second development iteration with the development team to present the RET profile obtained so far and the prescribed improvement practices. The main intention of the focus group was to gauge the practitioners' views on the suitability of the suggested practices. Furthermore, the relevance and validity of the presented distances and gaps was also assessed at the focus group (reported in Bjarnason and Sharp 2015). The whole team was invited to the session and six of nine team members attended. The content of the focus group was later covered with the three absent team members by individual semi-structured interviews. The focus group session and the interviews were audio-recorded.

The focus group began with an introduction to the Gap Finder and the concept of distances within software development. An overview of the full set of measured distances was then given before presenting and discussing the findings for each type of distance. First the obtained measurements were presented and an open question asked if and how this may have an impact on their work. The practices prescribed by the Gap Finder were then presented, and the team asked to comment on whether the practices may mitigate the distances and improve alignment, and if they were suitable to adopt.

After having discussed each type of distance, the participants were asked to reflect individually on issues related to the presented distances and practices for mitigating these. These reflections were written on post-it notes and then shared and discussed within the group.

At the end of the focus group the participants were asked if and in which way the Gap Finder process and outcomes were useful, including how they had experienced the workshop.

The audio recordings were transcribed and summarised. This summary was then distributed to all the team members who were asked to provide feedback if anything was incorrectly described or if they had additional reflections.

6.2.6 Survey of Prescribed Practices

The relevance of the practices prescribed by Gap Finder was further validated through a survey after the second development iteration was completed. The survey was sent by e-mail to one of the managers within the IT development unit, the product owner, the scrum master and the tester from the assessed team. For each prescribed practice the respondents were asked to indicate if it was planned to be implemented and what effect it might have. The survey template is available on-line, see Bjarnason 2013.

6.2.7 Post-study Interviews

We performed interviews five years after the main study to investigate if there is any evidence of the practices suggested by the Gap Finder method having been introduced within the team or remaining within the Department currently. The original study team had been disbanded two years after the study was conducted. Sharp held two semi-structured post-study interviews. One with the lead product owner of the studied development team who still works within the organisation, and one with a researcher who recently performed a study of the same IT

department and is, thus familiar with current work practices. We focused these interviews on the suggested improvements practices, in particular the ones initially planned for implementation, see Table 4. The respondents' viewpoints were captured in writing and confirmed by the interviewees.

6.3 Evaluation: Data Analysis

After completing the evaluation, the complete set of data gathered from applying the method and from the additional validation activities was analysed. The transcripts were coded by the first author and data from the different sources was analysed together for each distance, for each prescribed practice (RQ1) and for improvements (RQ2). The outcome of this analysis was discussed extensively with the second author. We thus analysed, compared and triangulated the full set of data for each research question.

The data was analysed in two main iterations, one before and one after the focus group session in which the initial findings were presented. During the initial analysis, the measurements, transcription of the interviews held in connection with the measurement surveys were stored in a spreadsheet and categorized (or coded) per distance type. For each type of distance, the relevant data was analysed together and compared against the observations. The aim of this analysis was to identify troublesome distances (gaps) and issues experienced within the development team, and potential connections between these. These findings were presented at the focus group session.

In the second round of data analysis the data on improvement practices were included in the analysis; from the focus group and the survey on practices. The focus group session and the interviews held in connection with the survey were transcribed and these transcripts were coded according to distance type and practice. The full set of data, i.e. measurements, profiling interviews, focus group, survey on practices, was analysed by considering all data relevant for each distance type and each improvement practice, thereby providing triangulation and validation of the outcome of the initial analysis. In addition, the participants' viewpoints on the Gap Finder method, the concept of distance and the suggested practices were coded and analysed together with the researchers' experience of applying the Gap Finder method. Finally, the data from the post-study interviews was analysed, compared to, and integrated with the results.

7 Results

The main results consist of the outcomes of applying the Gap Finder, namely the identified gaps and the prescribed practices, and the findings related to evaluating the method. We now present these results based on the data captured through observations, interviews, focus group session and survey.

7.1 Identified Gaps

Several troublesome distances, or gaps, between people were identified through the analysis of the RET profile, which was obtained based on the interviews held in connection with the measurement surveys. For the artefact-related distances only shorter (or no) distances were measured and, thus no artefact-related gaps were identified. The RET profile including the obtained measurements for each distance is outlined in Table 1.

Table 4 Overview of findings on relevance of prescribed practices including addressed distance, impact on requirements-test alignment, if team planned to implement the practice, and situation five years later

Practice	Addressed distance(s)	Impact on Requirements-Test alignment	Implementation planned	Post-study findings
P1 Guest desk	Geographical, cognitive	Improved requirements communication and validation	Yes, new	Implemented by original team, but impractical due to multi-project environment. Replaced by structured daily visits by Product Owner to the development team area and a commitment to stay as long as it took to resolve any issues.
P2 Requirements communication, all levels & throughout	Organisational, cognitive	Improved requirements communication and validation, decrease requirements conflicts and no of defects	Yes, new + strengthen existing practice	In the original team, this practice became more consistent and all-inclusive. Current study found this communication present in the organisation.
P3 Test cases reviewed against requirements	Semantic, cognitive, organisational	Ensure that requirements and test case artefacts are in synch, agreement of requirements	Maybe, if test department agree	Applied at organisational level, less so for original development team due to changes in role composition.
P4 Individual say in team seating	Psychological	None found. General impact on communication	No, space limitations	
P5 Product owner testing	Cognitive: domain, adherence	Improved verification of requirements	Yes, strengthen existing practice	Applied in original team and found in current organisation
P6 Competence development	Cognitive: technical skill	None found. General impact on abilities.	No, infeasible in practice	
P7 Job rotation	Cognitive	Improve alignment with system-test team by increasing their knowledge of agreed requirements	No, infeasible to apply systematically	
P8 Consider quality in elicitation	Cognitive	Early alignment on quality requirements can reduce the amount of late issues	Yes, new	Applied in original team, no evidence found for current organisation.
P9 Agee on quality priority for project	Cognitive	Increase agreement within team and organisation on quality requirements	Maybe, if rest of project agree	

7.1.1 Geographical Gaps (M1)

There was a gap in geographical location between certain team members. While the core team members, i.e. scrum master, developers and testers, were co-located in one common team area the product owner, requirements analyst and project manager were located elsewhere. The project manager had a desk in the same office as the team while the requirements analyst was located on a different floor in the same building. In addition, the product owner was located in a separate building approximately 322 metres away.

The team was aware of the negative impact of these gaps and frequently commented on the lack of proximity to the product owner and the requirements analyst both during the observations and at the focus group. This distance was experienced as causing time delays in obtaining requirements information, sometimes resulting in lack of communication or in obtaining this information via other potentially less reliable sources. The requirements analyst stated that the geographical gap towards the team reduced the frequency of direct communication and thereby also the general awareness of requirements within the team. The analyst attempted to mitigate this, partly through documentation.

7.1.2 Organisational Gaps (M2)

There was an organisational gap between the product owner and the other project members who belonged to the IT unit. The product owner was from outside of the IT unit at an organisational distance of 7 steps in total up and down the organisational tree. This was out of necessity since the role of the product owner is to represent the users, in this case the business owners.

Several team members described during the interviews that this organisational gap causes delays in decision making, and practical issues with coordinating meeting schedules were observed. The product owner mentioned that people from the business unit and those from the IT unit can disagree due to different priorities and perspectives, e.g. on how and which requirements to implement. In addition, various meetings at the IT department often conflict with other meetings held within the business unit causing scheduling difficulties for the product owner.

7.1.3 Psychological Gaps (M3)

Psychological gaps were identified between some project members. Two out nine members found it *Extremely hard* to communicate with one specific person, while three people reported that it required *Much effort* to communicate with two other project members. This was surprising since the team was perceived by themselves, within their organisation and by us as researchers as being well-functioning with good and open internal communication. This general perception corresponded well to the measured short average psychological distance; between *Not hard* and *Some effort required* to communicate with individual project members.

When shown these measurements at the focus group, the team members could recognize that this may explain some communication difficulties that had been experienced but not openly discussed due to their unspecific nature. For example, discussions about requirements were sometimes very polite rather than being open and frank. In addition, on a couple of occasions team members were observed to physically indicate reluctance to continue a discussion with a neighbour by turning away from the other person and focusing their attention on their screen thereby withdrawing from the conversation.

7.1.4 Cognitive Gaps (M4)

Within the team, cognitive gaps between team members with different roles and length of experience were found. Large distances were found for domain knowledge (M4.1) and knowledge of the local processes and organisation (M4.3) between team members who had joined the case organisation within the past year versus those who had worked there for several years. One of the developers pointed out this distance between the very experienced requirements analyst and the newer developers and tester during an interview. This developer related how this gap on several occasions had resulted in failure to identify incorrect software behaviour, e.g. through testing, due to the requirements analysts not communicating to the development team what he/she considered to be tacit requirements. These tacit requirements had thus not been developed or tested, and had not been discovered until later, during user acceptance testing.

Two surprising gaps were found for the aspect of prioritisation of system quality factors (M4.4). Firstly between the product owner and the requirements analyst who prioritized functionality and maintainability differently, and secondly for the tester who prioritised usability much lower than all other non-development roles.

7.2 Prescribed Practices

Nine practices (P1-P9) were prescribed as improvements based on identified gaps. These practices were presented and discussed with the development team at the focus group session. In addition, a survey on the suitability of the prescribed practices was sent to three team members and one manager within the organisation. The data collected for each of the prescribed practices is summarised in 0 and presented below together with a description of each practice and the researchers' reflections. The development team have indicated that they plan to implement four of the practices (P1, P2, P5 and P8), while two of them might be implemented (P3 and P9), and three are judged as infeasible to implement (P4, P6 and P7). In addition, three further improvement practices mentioned at the focus group by the team are reported.

7.2.1 Guest Desk for Product Owner (P1)

Practice Provide a guest desk in the team area for non-co-located team members, in particular the product owner.

Addressed distance Geographical (D1), cognitive (D4). The practice can bridge the distance by bringing the product owner physically closer to the team more often and for longer periods of time. This increased co-location can in turn bridge cognitive distance, in particular for domain knowledge, between the product owner and the development team.

Team response This practice was immediately picked up by the project team to be implemented. Even though this practice had been considered by the product owner before, it had not previously been discussed within the team. At the focus group, the scrum master said: 'Having the product owner in our office even 1 hour per day would help a lot with communication.' The product owner expressed that having a guest desk would enable working in between meetings rather than just waiting or spending time on walking back and forth. One team member believed that a guest desk would make the product owner feel more welcome and

encourage spending more time with the team, thus making this important role more available to the team. In addition, another team member believed that this could lead to an increased awareness for the product owner and enable this role to have more insight into the development team and the issues they face. However, office space is limited so reserving a desk for visitors is a challenge. In the meantime, an agreement has been made that the product owner spends more time in the team area and borrows temporarily available desks.

Reflections Measuring and visualising this known distance pinpointed it as a main cause of several observed communication difficulties within the project team and triggered the project team to immediately take action to implement the prescribed practice. Furthermore, the theory-based approach of the Gap Finder was observed to provide the project with sound arguments for requesting and subsequently obtaining the resources, i.e. additional office space, required to implement the change.

7.2.2 Requirements Communication at all Levels & Throughout Project Life-Cycle (P2)

Practice Establish additional and strengthen existing communication paths from the team to roles and functions currently with insufficient requirements information, e.g. for the system test team, between tester and product owner.

Addressed Distances Organisational (D2) and cognitive (D3). The practice can mitigate organisational distance by creating short-cuts in situations in which increased requirements communication may avoid later misunderstandings, e.g. between product owner and tester, or between requirements analyst and team-external testers. Requirements validation can also be improved by bringing together roles with different knowledge and perspectives. In addition, the practice can affect cognitive distance by bridging it in the short term, and decreasing it in the long term by sharing knowledge.

Team response At the focus group, a majority of the team members were positive about this practice and even though the practice was already applied by the team, they stated that it was desirable to further improve on their communication practices. One of the survey respondents said that an increase in requirements communication would reduce the number of unpleasant surprises that surface later on, e.g. issue reports, and contribute to a better understanding of different viewpoints. This in turn would result in fewer disagreements around requirements. Similarly, a developer suggested that by involving the user interaction team in requirements discussions the interaction designers would gain ‘a better appreciation of why we ask for particular things and why we think they are important’. This would contribute to decreasing the number of disagreements between team roles has and those at an organisational distance.

The developers commented that it may be possible to have more frequent demonstrations of on-going requirements work throughout the sprints. However, they believed that the product owner’s limited availability and physical presence in the team area would restrict the frequency of this practice.

One focus group participant reflected that the communication between the product owner and the (current) tester could be increased. A previous tester had been very interactive with the product owner in showing mock-ups and discussing requirements details.

Another participant mentioned that establishing additional communication paths between the team and roles from other organisational units, e.g. to the system testers, could decrease disagreements and the number of system-level issues.

Reflections Even though this practice was already applied for the case project, the identification of specific gaps, e.g. between the product owner and the tester pinpointed additional communication paths for improvement. This in turn triggered the focus group participants to propose more specific practices for improving these paths, e.g. demonstrations.

This practice demonstrates that there are relationships between distances and thus identifies an area for future research. In this case, we see a correlation between organisational and cognitive distances, e.g. between product owner and tester. Increased direct communication between roles from different units bridges the organisational distances and over time reduces the cognitive distances because there is increased sharing of information. Over time, this sharing aligns the knowledge of those involved.

7.2.3 Test Cases Reviewed Against Requirements (P3)

Practice Let someone other than the tester, e.g. the requirements analyst, look at the test cases and consider if they cover and correspond to the requirements in an adequate way.

Addressed Distances Semantic (D6), cognitive (D4), and organisational (D2). This practice primarily reduces semantic distance between artefacts and cognitive distance between the roles responsible for those artefacts, and may also bridge organisational distance between these roles. The semantic distance is reduced by identifying and remedying the causes of the gaps through the review and subsequent updates. The cognitive distance between the reviewers can be decreased by the information shared during the review. Potential organisational distance between the involved roles (which for this case is not an issue) can be bridged by the communication channel set up by the review practice.

Team Response There was a mixed response to this practice although focus group participants confirmed that it can have an impact on requirements-test alignment. One participant said that having more people look at the test cases would likely result in improving them. Another said that the practice would increase the sharing of knowledge concerning test cases. Furthermore, one survey respondent believed that this practice would support the requirements analysts in writing clearer and better acceptance criteria. Two of the survey respondents stated that this was a practice to adopt. Another survey respondent was less definite and said that the practice might be useful. A possible explanation to these responses could be that the decision on implementing this practice lies with the testing department, and not with the team itself. However, the tester who belongs to this department expressed no opinion about this practice.

Reflections The responses to this practice demonstrate the importance of presenting the prescribed practices to those responsible for the connected process areas. Other roles may have opinions about them, but ultimately it is the responsible team/unit that needs to assess and make the decision regarding adopting the practice.

7.2.4 Let People Have a Say in Seating Arrangements (P4)

Practice Let personal preferences regarding ease of communication be one factor when considering team seating.

Addressed Distance Psychological (D3). This practice can bridge distance by allowing people some control over their communication with others.

Team Response At the focus group, the scrum master indicated that psychological distance could be one factor among many to consider when deciding on how to locate different team members. However, in general the response on this practice was that it is hard to accommodate since office space is limited and opinions vary. This practice triggered another focus group participant to initiate a discussion on whether seating people next to each other between whom there is a long psychological distance might decrease the distance, or alternatively decrease the communication.

Reflections There are multiple aspects to consider in influencing the frequency of communication between different individuals. Apart from psychological distance, factors such as cognitive distance and the importance of frequent communication between different roles are likely to have an impact. However, it is unclear how to balance and optimise these different factors. Further research is required to provide evidence-based guidelines and recommendations.

7.2.5 Product Owner Testing (P5)

Practice The product owner or the requirements analyst performs user testing with the intention of validating that the implemented behaviour and performance aligns with overall system intentions and user expectations.

Addressed Distance Cognitive (D4), in particular concerning domain knowledge (M4.1), and adherence (D5). Differences in domain knowledge between the one performing the product owner testing and the tester in the development team can be bridged by utilising their additional knowledge of the user requirements and the domain. The requirements validation supported by this practice can thus decrease the adherence distance between the agreed requirements and the implemented software behaviour by identifying and addressing mis-alignment within the development team.

Team response This practice was already applied by the product owner and the requirements analyst, who at the focus group both indicated that they would consider applying this practice more often. Similarly, the scrum master stated that it is a practice that the team will apply more often in the future. The focus group participants agreed that the practice strengthens requirements-test alignment through validating that the developers and testers have correctly understood and fulfilled the customer requirements. The practice has uncovered issues related to missed or misunderstood requirements details and as one survey respondents said can ‘help to highlight problems with misunderstandings and wrong assumptions earlier in the process, and help the [requirements] analyst feel closer

to the working software.' However, technical limitations were also mentioned in that the product owner cannot access the software for the sprint until after it has been delivered, thus only allowing the product owner to apply this practice after the team has delivered. The requirements analyst however was observed to perform this testing during a sprint, thereby identifying a number of missed requirements.

Reflections The researcher applying the Gap Finder was unaware of the existing use of this practice prior to the focus group. An upfront inventory of existing practices may further improve and sharpen the prescription step to identify changes to the current practices including improvements to how they are implemented.

7.2.6 Continuous Competence Development (P6)

Practice Increase team member's technical knowledge through personal study, training courses etc. within specific areas, e.g. testing.

Addressed Distance Cognitive (D4) and, in particular Technical skill (M4.1). Decrease cognitive distance by increasing the skill level of individual members.

Team Response While there was agreement in principle to this practice, the practice was not seen as feasible to adopt in a systematic way. One survey respondent expressed that increased competence in teams and individuals would increase their ability to adapt and deal with challenging situations and thus indirect influence requirements-test alignment. At the focus group, one team member commented that the majority of the team members were short-term contractors and indicated that competence development was mainly considered for permanent staff.

Reflections Since the effects of this practice are not immediate, it is likely to be more suitable for organisations with a large percentage of long-term employees and a strategy and plan for learning and competence development. For example, a competence programme with defined categories and levels of competence enables an open and objective discussion on which level each person is at and what is required for different roles and position. The existence of a gap in competence for each individual, but also at the overall level, can then be identified and training programmes be defined to address these. Furthermore, the decision to implement this practice would need to be made by the line manager rather than the project organisation.

7.2.7 Job Rotation (P7)

Practice Rotate team members to different roles and responsibilities, e.g. team tester to system test team, requirements analyst to testing.

Addressed Distance Cognitive regarding aspects technical skill, organisational & process knowledge. This practice primarily addresses the organisational and process knowledge aspect of the cognitive distance by increasing a person's knowledge of a new role. In addition, this

person can gain technical skills by performing another job, thus also decreasing the distance for the technical skill aspect.

Team Response The focus group participants expressed that it would be challenging to apply this practice even though gains had been observed when it was applied in an ad hoc fashion. For example, when a team tester had been transferred to the system test team. One survey respondent stated that the practice would incur additional costs in the form of a temporary productivity drop and increased training needs. In particular, several focus group participants mentioned that it would be hard to handle the loss of competence caused by rotating a team member. Furthermore, as indicated by another focus group participant, rotating to a different role may not be in-line with personal preferences. As expressed by one survey respondent, the practice would cause 'improved general knowledge of different areas, but at a cost of less specific knowledge.' However, an interviewee described that it was not uncommon for people to be moved between teams and roles as the need arose. In particular, since most of the requirements analysts also have system testing experience it had been discussed within the organisation to have the analysts take on testing roles.

Reflections This practice is similar to the previous practice, although it concerns practical competence development (as opposed to gaining theoretical knowledge). Similarly, this practice is relevant for line managers of development organisations to consider when assessing competency needs overall, and thus illustrates that requirements-test alignment is not merely a project-level concern, but also something to consider at the organisational level.

7.2.8 Consider Quality Upfront in Requirements Elicitation (P8)

Practice Consider quality characteristics during the requirements elicitation by identifying important quality requirements in the early discussions. These are then detailed in the same way as other requirements.

Addressed distance Cognitive (D4), in particular the priority of quality aspects (M4.4). Cognitive distance concerning the priority of quality aspects can be decreased or (at least) bridged by discussing and sharing different perspectives on their relative important.

Team Response Several team members stated that this is a practice that they plan to adopt since it will help identify quality requirements early on, which can reduce the number of issues discovered in later activities, e.g. systems integration and production, and thereby avoid costly and time consuming maintenance. As one survey participant said: this practice 'might catch particular issues earlier when they are easier to address.' However, at the focus group some team members expressed doubt whether quality aspects could be elicited upfront since there might not be sufficient awareness of the customers' expectations for these aspects.

Reflections The mentioned difficulties of defining quality (non-functional) requirements at an early stage in the development cycle demonstrate that the prescribed practice poses requirements on another cognitive aspect, namely knowledge of the domain and the customer's

expectations. This also illustrates the existence of pre-requisites for a practice to be practically implementable, i.e. certain distances may be required to be short for a practice to be suitable.

This also illustrates that temporal aspects of the development process influence the suitability of practices. On one hand, defining quality requirements upfront (P8) may mitigate cognitive distances by eliciting clear requirements at an early stage thus reducing misunderstandings and subsequent software errors. On the other hand, end users may not have the insight required to identify what their requirements are at this early stage. Thus they would benefit from several shorter cycles between defining and implementing requirements in order to gradually gain the insight necessary to identify their actual requirements. This indicates that there are complex relationships between distances that need to be considered when prescribing practices.

Defining quality requirements are a known challenge for software development (Berntsson Svensson et al. 2009), in particular for agile development (Ramesh and Cao 2010), but also for traditional development models. These aspects are often not explicitly defined as requirements and instead result in issues with the delivered software. For example, during user testing when the performance or the capacity of the system are unsatisfactory from a user perspective, i.e. issues that are often costly to address. New insight may be gained by applying the concept of distances to this challenge and in particular, how to manage the relationship between cognitive and temporal distances when balancing the timely definition of requirements against the maturity required to defined correct and feasible requirements.

7.2.9 Agree on Quality Priority for Project (P9)

Practice Discuss which quality aspects are more or less important for the project and establish a common view of relevant quality characteristics within the team.

Addressed distance Cognitive (D4), in particular the priority of quality aspects (M4.4). Agreeing to a set of quality aspects for the project can decrease the priority aspect of cognitive distance between team members because they will have shared and aligned their various viewpoints.

Team Response At the focus group, this practice triggered a discussion on how quality is perceived differently depending on role and responsibility, thereby illustrating the need for explicitly agreeing on these. The practice was stated by three of the four survey participants as one to adopt. One of them said that the practice 'could help us come to a more common understanding of where we should be focusing our efforts', i.e. support the alignment of quality requirements at the project level by defining goal-level requirements for quality. However, the mandate for deciding to implement this practice lies with the project rather than with the development team.

Reflections Since this practice mitigates the (known) challenge of quality requirements at the goal level (a higher abstraction level than for the previous practice related to quality requirements, i.e. P8) it can be expected to have an impact also on requirements-test alignment for the higher levels of testing, e.g. system and user acceptance testing. While some of the temporal challenges with defining user requirements early on may still apply, we believe it is a more

manageable challenge at this higher level since less detailed knowledge is required to prioritize general quality attributes than to define actual quality requirements.

The response from the project team indicates that the prescribed practices need to be presented to those responsible for processes affected by the suggested practices. For example, the decision to introduce a project-wide agreement on priority of quality needs to be made by management for a project or for the entire IT-department.

7.2.10 Additional Practices Suggested by the Team

In addition to reflecting on the prescribed practices, the focus group participants suggested the following practices for mitigating the found gaps:

- The scrum master suggested that misunderstandings of requirements caused, e.g. by organisational distance (D2) could be decreased by improving the acceptance criteria so that they become more like acceptance test cases. Thus, decreasing the semantic distance (D6) between the requirements artefacts and the test cases (M6.1). This suggestion corresponds to the practice of using test cases as requirements specification, covered by AP4 (Aligning documentation).
- One developer suggested that geographical (D1) and psychological (D3) distance could be further shortened by re-organising the team area thereby further improving communication including requirements clarification and detailing. For example, removing dividing screens and placing desks facing each other rather than back-to-back would further facilitate visual contact and awareness. Furthermore, this team member mentioned implementing additional communication practices, e.g. always face the person you are talking to, listen when others are talking. This suggestion is related to the AP1 practices of cross-role collaboration.
- One developer suggested that cognitive distance (D4) concerning technical skill (M4.2) could be decreased by ensuring that all team members are able to access each other's artefacts. For example, no other team members currently have any knowledge of the test cases and cannot access or view them. Similarly, there is a lack of access to other artefacts produced by previous team members, e.g. requirements documents for previous sprints. This suggestion is related to the AP1 practices of cross-role collaboration.

7.3 Practitioners' View on Gap Finder

Throughout the study feedback was gathered from the team members concerning their experience of the Gap Finder, concerning both its approach and output, and the time and effort required from them to participate in the assessment. The main feedback was gathered at the focus group, but also as part of the interviews and through the observations.

At the focus group, the team members expressed that they found the approach of the Gap Finder useful in discussing issues and in identifying new areas for improvement, and that the suggested practices were appropriate. Even though several of the suggested practices were not completely new to the team, e.g. guest desk, product owner testing, presenting and motivating them in light of the concept of distance provided additional motivation for deciding to implement them. The fact that the prescriptions were derived in a structured and theory-

based manner provided a credible and viable argument in obtaining the resources required to implement some of the practices. For example, based on the suggestion of the Gap Finder to have a guest desk for the product owner (practice P1), the development team were successful in acquiring an additional desk for this purpose, something that was otherwise hard to obtain since free office space in the development area was very limited.

Furthermore, the focus group revealed that the product owner, who usually does not attend the team retrospectives, was in fact more positive towards adopting some of the practices than the rest of the team thought. In particular, this was the case for P1 Guest desk for product owner. In addition, some of the suggested practices were already applied by the product owner although the team was not aware of this, e.g. P5 Product owner testing.

The development team did not find the assessment particularly costly from their perspective. Even though they had a high work load they found time for answering the questionnaires, which could each be done in 10-15 minutes. When asked, the scrum master also expressed that the team had not perceived any undue cost of participating in the evaluation study. The most time consuming part from their perspective was the final Gap Finder workshop, which took just over 60 minutes. The researchers' experience was that the workshop required longer time than what was available. To reflect on and discuss all the different distances in a satisfactory way required the participants to take in a lot of information.

7.4 Evidence of Practices Being Applied After This Study

We explored evidence of the practices being applied by members of the studied team and the IT department through post-study interviews with the lead product owner from the original team, and with a researcher recently studying the IT department. There have been many changes in the IT department during the five years since the Gap Finder was applied, including that the original team was disbanded 3 years after our study. However the product owner shared that after the Gap Finder study, communication overall improved and the team's velocity increased, and that the team delivered working software that is still in use today. The product owner explained that since practices were evolving, and teams were consolidating and becoming increasingly familiar with the code and the requirements, it is not possible to say exactly what caused these improvements in requirements communication.

For the five practices planned for implementation, our interviewees indicated that these practices or variants of them were implemented by the studied development team and/or remain part of current practice today. An overview of the status of the suggested improvement practices is shown in Table 4.

For *practice P1 Guest desk*, a hot desk was allocated for use by the product owner, but this was not an efficient way to work when the development project was only one of many work tasks. Thus, the multi-project work environment of the product owner made this practice inefficient. Instead, the product owners would visit the development team for the stand-up meeting each day and remain to discuss and resolve any issues that had arisen. This practice persisted until the development team was disbanded.

The product owner stated that practices *R2 Requirements communication at all levels* and *R5 Product owner testing* were implemented, and the interviewed researcher described that today these are part of common practice within the IT department. The product owner described that increases in communication (P2) led to more reuse of source code and thus the velocity of the development team was increased. In addition, product owners performed user acceptance testing (P5) including producing test plans and schedules, and with this approach testing went well and

found issues were addressed. The researcher interviewee stated that *product owner testing* (P5) is now part of the current development process, although this increases the workload on the product owners and causes subsequent delays in the affected process step.

The practice P3 *Test cases reviewed against requirements*, was mentioned by the interviewed researcher as part of current practice. The product owner described that the developers had taken on the responsibility for the unit and integration testing when their tester was no longer assigned to the team, which led to a more efficient process in which the product owner communicated with the same individuals for both implementation and testing.

Finally, the product owner stated that the practice P8 *Consider quality in elicitation* had been implemented as suggested by the Gap Finder. This led to discussing quality issues such as usability, accuracy and efficiency early on in development. The interviewed researcher had not observed this practice in the current organisation.

8 Findings and Discussions

Our study provides experience of applying the method and yields a number of new insights concerning how the Gap Finder method can support a project team in improving their requirements communication and requirements-test alignment, as well as insight into how the method itself can be improved. Based on the results presented in Section 7, we can now answer our two research questions regarding the relevance of the Gap Finder prescriptions and how the method can be improved. We also discuss the value of the method from the perspective of the development team, and the validity and limitations of our study.

8.1 Relevance of Prescribed Practices (RQ1)

Our evaluation showed that the majority of the prescribed practices were relevant vis-à-vis the detected gaps and for the project for which the Gap Finder was applied. The project team stated that seven of the nine practices prescribed by the Gap Finder could directly improve their communication by addressing requirements-test alignment for their project. Furthermore, our study triggered action and one of the prescribed practices in particular, namely Guest desk for product owner (P1), led to immediate action to implement despite limited free office space in the IT department. The objective and theory-based approach of measuring distances and prescribing improvements based on empirical evidence provided the project team with factual arguments and motivation to request the necessary resources to implement these practices. However, the cost-benefit balance varies between practices and thus also the feasibility of implementing them.

At the time of our study, the team planned to implement four of the practices (P1, P2, P5 and P8). Two practices might be implemented (P3 and P9), while three (P4, P6 and P7) were judged as infeasible to implement. Two of the prescribed practices (P2 requirements communication and P5 product owner testing) were in fact already applied in the project, at least partially. For product owner testing (P5), not everyone was aware of the use of the practice. All three of these practices were planned for further and improved implementation. Thus, the team benefited from including these in the Gap Finder prescription despite them already being applied.

Three of the prescribed practices were generic and less specific than the others, and concerned cross-role collaboration (P2), seating arrangements (P4) and job rotation (P7). For cross-role collaboration (P2), specific communication paths to improve, e.g. between product

owner and tester, were discussed at the Gap Finder workshop based on the identified gaps and the information in the RET profile. This illustrates how group reflections can identify and agree on improvements, and thus validates the group reflection element of the Gap Finder. However, the specificity of the prescribed practices may improve the relevance of the prescription and is a possible area for improvement of Gap Finder, see Section 8.2.3. More specific improvement suggestion could provide more value to the project team.

The other two generic practices, team seating (P4) and competence development (P6), were not included in the improvement plan because they were not expected to impact directly on the project's communication. The project team believed that these practices have a general impact on communication and abilities within the team that can be expected to be seen over time and thus not limited to a specific project. This could explain the project team's rejection of them. Rather, these practices are relevant for line managers to consider when defining an organisation's strategy and plan for competence development.

The long-term effects of the Gap Finder are hard to pinpoint decisively due to many changes within the IT department beyond those incurred by applying our method during the five years since our study. However, based on the information we have gathered, the practices were implemented by the team and their performance improved after our study. In addition, several of the practices appear to be part of current practice in the IT department, namely P2 *Requirements communication at all levels and throughout*, P3 *Test cases reviewed against requirements*, and P5 *Product owner testing*.

We found evidence that three of the suggested practices (P2, P5 and P8) were successfully implemented in the studied team, while two of the practices (P1 and P3) had been adjusted to achieve a better fit. Of these, the practice P1 *Guest desk* was implemented but the product owner's additional work responsibilities beyond the development project made it infeasible. Thus, the multi-project work environment of the product owner role affected the efficiency of the suggested process. Instead, the product owner mitigated the underlying geographical and cognitive distances by allocating specific time, around the stand-ups each day, to be physically present and available to the development team, thereby leading to a closer working relationship and improved requirements communication. From this, we conclude that factors external to the project, and thus not investigated, affect the accuracy of the prescription.

The other practice that was adjusted was P3 *Test cases reviewed against requirements.* Changes to the team composition, namely shifting the testing responsibility to the developers resulted in a change in the underlying communication paths, replacing the product owner's distance towards a tester by the distance towards the developers. The product owner's experience that this is a more efficient process aligns with the shortened and more frequently traversed distance over this communication path, compared to the one to the tester. We observe that the concept of distance can be used to explain and support both of the adjustments in practice.

8.2 Improved Usefulness of the Gap Finder (RQ2)

While the development team and several IT-managers expressed that participation in this formative evaluation of the Gap Finder provided the organisation with value by yielding new insights and perspectives, we also identified a number of potential improvements to further enhance the usefulness of the Gap Finder. These improvements can be divided into four areas namely a) the set of distances to measure, b) the identification of improvement practices, c) how to support team reflections at the Gap Finder workshop and d) strengthening the software process improvement aspects of the method. We will first discuss and compare the benefits and

the costs of applying the Gap Finder, and then outline suggestions for improving the effectiveness of the Gap Finder method for each of the areas a-d.

8.2.1 Costs versus Benefits

The *costs* of applying the Gap Finder consist of the work required for a moderator to prepare and perform the assessment, and that of the development team being assessed. For our case study, the development team including its Scrum master stated that the time required of them to apply Gap Finder was modest and did not negatively affect their day-to-day work. This also included the additional data collection required for the evaluation, i.e. interviews, practice surveys, and additional focus group questions. The main cost to applying the method lies in the work required for the moderator to plan and gather the measurements, and to perform the prescription step to identify communication gaps and improvement practices. To some extent, this time can be reduced as the method matures and is improved, but will require that a moderator is assigned to this task. An interesting avenue for future research is how to measure relevant distances more efficiently, e.g. by instrumenting existing development environments and tool chains.

We found that the main *benefits* of the Gap Finder method are providing evidence-based improvement suggestions, novel explanations for communication gaps, and support for discussing communication issues, both in general and for sensitive issues, in an objective manner. Firstly, the *theory-based approach* used to identify practices for improving requirements communication provided the development team with fact- and evidence-based arguments for implementing new practices. Some of these had previously been discussed but not pursued due to a lack of management support prior to being suggested by the Gap Finder. The practices proposed by the Gap Finder appear to be largely relevant, although we also identify improvements that may increase the accuracy and efficiency of the method, see below.

Secondly, the concept of distance provided *alternative explanations* to previously discussed communication gaps. For example, at the Gap Finder workshop the team gained new insight into why disagreements between organisationally distant people caused difficulties and long lead times to resolve. One mentioned example of this was when there were disagreements about the scope between the Product Owner and the rest of the team. Seeing distance as an explanation for the communication gap, triggered the Product Owner to reflect on the fact that her strategies to manage these situations was to bridge and shorten the organisational and geographical distance, through regularly attending meetings at the IT department even though this was not officially part of her assignment.

Thirdly, the Gap Finder provided the team with a forum for discussing and sharing viewpoints on *communication issues*, including sensitive issues. At the focus group, the team as a whole gained new insights into each other's preferences and ways of working. For example, the product owner shared that she occasionally performed end-user testing, something the developers were previously unaware of. Furthermore, the distance measures enabled members to discuss communication gaps in an objective fashion. In particular, the measurement of psychological distance triggered an objective and constructive discussion at the Gap Finder workshop of previously undiscussed communication gaps between team members. These discussions led to revealing differences of opinion regarding work practices, caused by different previous experiences and personal preferences.

Finally, the improvements identified for the Gap Finder (see next Section) can reduce the time and effort required for the method. This includes focusing the set of distances and improvement practices discussed at the Gap Finder workshop, and referring decisions on implementing improvement practices to forums with relevant decision mandates.

8.2.2 The Set of Distances

Even though most of the distance measures we used were found to be relevant, some were less so. In particular, for the cognitive aspects of technical skill and organisational and process knowledge no evidence of impact on requirements communication was found. Rather, at the focus group the participants described themselves as a well-functioning team consisting of members partly due to the range of competences and degree of seniority among the team members. Furthermore, two additional aspects were identified as relevant and useful when considering requirements-test alignment. These are, 1) a cognitive aspect concerning the difference in knowledge of agreed requirements, e.g. between a development team and the system testers, and 2) an adherence aspect between the abstraction level of agreed requirements and the behaviour of the delivered software. Both of these aspects are candidates for being added to the Gap Finder.

The surveys used to measure distances using self-assessment, e.g. for cognitive distance, pose a risk of incorrect, or biased measurements. This was observed to be the case in particular for the artefact distances. When asked to assess distance between related requirements and test cases the respondents' initial response was to state no distance, even for cases in which the test cases verified an agreed but non-documented change to the requirements. We believe this risk of bias is due to the fact that the existence of a distance in similarity or coverage between the documented and/or delivered requirements or test cases correlates to a failure to capture and match the agreed requirements. The person responsible for ensuring that an artefact has sufficient similarity and coverage might be unable or unwilling to detect a distance. For this reason, alternative measuring approaches may need to be investigated, especially for the adherence distance between agreed and documented requirements and the semantic distance between requirements and test artefacts. For people-related distances, such as cognitive and psychological distance cost-effective alternatives to self-assessment are harder to envision, but still remain an interesting avenue for future research.

Finally, there is potential for improving the efficiency of the Gap Finder method by investigating which set of distances can facilitate identifying the most significant and serious communications gaps within a development project, i.e. the communication issues for which improvements can bring the biggest gains. Identifying such a set would also reduce the cost of measuring and analysing distances by focusing on the most cost-effective distances.

8.2.3 Prescribing Improvement Practices

We identify several potential improvements in this area. First, some of the prescribed practices were in fact already applied, although this had not been caught during the observations or the interviews. The Gap Finder prescription could be improved by adding a step for identifying

existing practices, e.g. through a survey, prior to the analysis. This information can then be considered during the analysis, but should not exclude the possibility of suggesting existing but relevant practices. Rather, for these practices, the outcome should be a suggestion to consider how to further improve on their implementation.

Second, the focus group participants expressed that some of the suggested practices may rather have an indirect affect on requirements-test alignment, i.e. team seating (P4) and competence development (P6), and their impact on distance needs to be investigated further. It may be that these practices need to be tailored further to mitigate specific gaps. For example, if a gap for a specific competence is found, then this technical area and the involved individuals should be suggested for competence development. For the practice of team seating and its associated psychological distance, further insight is needed either from literature studies or from additional research into the impact of this practice.

A third aspect to consider is how to identify the case-specific improvement practices based on the abstracted practices derived from the Gap Model. Currently this is done prior to the Gap Finder workshop. An alternative approach would be to present more high-level and generic improvement practices and then refine these at the workshop together with the development project. An additional, future improvement could be to extend the underlying theoretical model (i.e. the Gap Model) with specific practices suitable for different sets of communication paths, e.g. between product owner and tester.

In general, increasing the accuracy of the set of prescribed improvement practices can increase the effectiveness of the Gap Finder method. However, identifying the overall most effective set of practices is non-trivial, due to relationships and correlations between the practices and the distances. Furthermore, organisational, and personal characteristics and preferences also affect the effectiveness when implementing a new practice. For example, the use of a guest desk was not feasible for the product owner due to other project-external factors that were not considered in the analysis. For this reason, we believe that insight into the specific organisation, and involvement of this organisation in the process improvement effort is vital to ensure a good fit of practices and buy-in for implementing them.

8.2.4 Workshop Set-Up

The Gap Finder workshop enabled the team to jointly reflect on issues and improvement practices. In particular, the concept of distance provided a good metaphor for supporting the team to discuss known issues from a new perspective, and to address previously un-discussed issues. However, these discussions could be improved by *focusing on key gaps* that are particularly relevant to the assessed case. This would ensure a more effective use of meeting time while also reducing the set of distances and measures the participants are required to consider.

The group reflections may become more efficient by further focusing the discussion and presenting distances grouped by relationships within the project and the combined set of distances for these. For example, all distances between the tester and the product owner, or between the development team and the system test team. The current presentation is organised by type of distance, even though several practices affect multiple distance types and there are connections between distances. However, discussing individual relationships might be sensitive even with objective data and is an aspect to consider here.

8.2.5 Software Process Improvement Aspects

The main aim of the Gap Finder is to support enhanced requirements-test alignment by suggesting practices that will improve RE integration with testing activities. This requires identifying suitable practices and supporting the team in deciding which practices to implement. For this study, application of the Gap Finder was not iterated, so the agreement about which practices to implement was gathered over a period of 4-6 weeks - mainly through a survey. However, it is desirable to reach an agreement that is committed to by the whole team since participation is a known major factor for successful software process improvement (Dybå 2005). The risk of failing to implement identified process improvements is implicitly mitigated by involving the whole project team at the workshop. Involving team members in reflecting on communication issues and improvements increases the chances of them later implementing the agreed practices.

In our study, decision making regarding which practices to implement was not included in the Gap Finder method as such but managed by the team itself after completing the main study. This delay in process improvement may be reduced by including the decision making step in the Gap Finder method. One way to gain agreement on which practices to implement could be to have two sessions of the workshop, i.e. one session similar to the existing one and one follow-on session for process improvement planning. The main aim of the second session would be to define and agree an action plan for implementing process improvements. This would also allow the participants to reflect individually before meeting again to agree on which improvements to implement.

For this case the decision to adopt two of the suggested practices (P6, P7) lay outside the team, e.g. with the line manager for the testers. This indicates that the mandate for implementing each of the prescribed practices needs to be taken into account by the method. These mandates could be identified at the first workshop session. The full set of stakeholders including the affected team members could then be invited to the second session focusing on process improvement planning. This would ensure sufficient mandate to decide on which practices to implement and agree to an improvement plan, and the decision process regarding implementing improvements may be more efficient by including this step in the Gap Finder.

8.3 Validity and Limitations

In this section, we discuss the limitations of this work according to guidelines suggested for design science research (Burstein and Gregor 1999) and for case studies (Runeson et al. 2012). Steps taken to mitigate these limitations and threats to validity are also mentioned.

8.3.1 Significance

The Gap Finder is a contribution of both practical and theoretical significance. The method addresses an industrially relevant challenge, namely that of aligning requirements and testing (Uusitalo et al. 2008, Bjarnason et al. 2013). Our evaluation shows that practitioners find the method helpful in identifying relevant improvement practices at a reasonable cost.

The theory-based approach taken by Gap Finder to address SPI in the area of requirements-test alignment is a novel contribution. There is some research on SPI methods for requirements-test (Kukkanen et al. 2009, Unterkalmsteiner et al. 2013, see Section 3.3) and on theory-based SPI approaches (Brede Moe et al. 2009, Angermo Ringstad et al. 2011, see

Section 3.4). However, we are not aware of any other research on theory-based SPI methods in general, or for requirements-test in particular.

Finally, the application of the Gap Finder contributes to testing the underlying theory and thereby provides empirical evidence that validates the theory. In addition, this data can be used to further develop the theory of distances (Bjarnason et al. 2016).

8.3.2 Internal Validity

The main threat to internal validity of the Gap Finder is the extent of qualitative analysis currently required to assess gaps and prescribe improvement practices. This risk is partly mitigated by the use of an underlying theory to guide the set of distances assessed and practices prescribed. However, further testing of the Gap Finder and development of the theory could provide more guidance in this analysis. For example, the theoretical model used to prescribe practices, i.e. the Gap Model, could be extended with parameters regarding the impact on specific communications paths and influencing contextual factors.

The main threat to internal validity for the evaluation of the Gap Finder is in misunderstanding or misinterpreting the empirical data. This risk was partly mitigated by collecting data from multiple sources, i.e. interviews, observations, surveys etc. In addition, all results, both intermediate and final, were identified by one researcher and reviewed by another.

8.3.3 External Validity

The question of external validity concerns the extent to which the outcome of this study are applicable and of interest beyond that of the studied case. For this analytical generalisation needs to be considered and the validity assessed on a case-by-case basis by comparing the specifics of a separate case alongside the full set of characteristics reported for this case (see Section 5). In addition, the generalisability is also limited by the scope of validity of the underlying theory and the external validity of the underlying theoretical model, i.e. the Gap Model.

The scope of validity for the theory of distances and the Gap Model is derived from the empirical data for the case study of five software development organisations in which these are grounded (Bjarnason et al. 2016). This scope represents the alignment of requirements and testing for cases within a range of case characteristics concerning organisation and project size, i.e. small to medium, project duration within0.5-5 years, process model ranging from traditional to pure agile), and safety criticality from high to medium degree.

We believe that the Gap Finder can be valid and valuable for cases similar to our case environment, namely small to medium-sized co-located software development organisations and projects with a project duration of 0.5-1 years and that apply an agile method. This range will be further extended over time as the Gap Finder and the underlying theory are tested and developed further, thereby extending their external validity. This includes knowledge of what comprises a troublesome gap in a RET profile, as well as more fine-grained rules concerning contextual factors that impact how a practice affects a distance.

8.3.4 Construct Validity

The main risk to construct validity is that the prescribed improvement practices were evaluated based on the project members' beliefs of their suitability rather than directly assessing how these practices would affect the project's communication. Our original plan was to implement

a set of suggested improvements practices and re-assess the distances. However, this direct assessment was not possible due to changes in the case project. This threat to construct validity was partly mitigated by evaluating the prescription through a focus group and through a survey with project members. The post-study interviews also provided some insight into the long-term effect of the Gap Finder. However, the iterative application and long-term effect of Gap Finder and subsequent re-measurement of distances remains as future work.

Another potential risk to construct validity concerns the appropriateness of the underlying theory, specifically in relationship to the external validity of the Gap Model relative the case used for the evaluation. However, we judge the cases as comparable when considering requirements-test alignment since the characteristics of the evaluated case (described in Section 5) fall within the range of case characteristics of the underlying theory that includes medium sized agile development projects. Furthermore, we have found no data in this study that conflicts with the underlying theory on which we have based the Gap Finder.

Finally, we provide a transparent and detailed description of our research method in order to support other researchers in validating our study.

8.3.5 Reliability

There is a risk that researcher biases have influenced the application and evaluation of the Gap Finder and, thus the reliability of the results of this study. This risk was partly mitigated by defining the aim of our research and clearly defining research questions to guide it.

The reliability of our research was further increased by including multiple researcher perspectives on design aspects at several points throughout the study and by applying triangulation to the collected data. For example, the research design and the data collection instruments were iteratively reviewed and refined by the researchers. Triangulation of the obtained distance measures was done by also collecting data on each distance through observations and interviews. Finally, the practices prescribed by Gap Finder were presented to and discussed with the development team in order to validate the suitability of each suggested practice.

9 Conclusions and Future Work

Software process improvement aims to improve the productivity of software development by identifying suitable improvement practices. We designed the Gap Finder method to target requirements communication within a development project by improving the alignment of requirements and testing. Our method assesses distances that can have an effect on requirements-test alignment and provides a prescription of practices for mitigating these underlying factors. The Gap Finder consists of a bottom-up SPI assessment (the measurement step), a prescription step in which case-specific improvement practices are selected based on an empirically based theory and a retrospective in which the assessment and the proposed improvements are reflected on and discussed within the project team.

In this paper, we report on the design and evaluation of the Gap Finder method. We have applied the design-science approach in designing our method based on existing scientific knowledge and theory, and evaluated it by applying the method to an ongoing development project. The evaluation shows that the Gap Finder can

provide relevant and useful suggestions for improving requirements communication (RQ1) for small to medium-sized co-located agile development projects. In addition, the Gap Finder approach was found to also provide evidence-based motivation for implementing some of the prescribed practices. While all of the practices prescribed by the Gap Finder can support improved communication, some target long-term and more general improvements. More precise improvement practices that pinpoint, e.g. specific communication problems, would provide more immediate value to a development project.

The insight gained through the case study enables us to identify several potential improvements to the Gap Finder method (RQ2) increasing its usefulness to a development project. The relevance of the prescribed improvement practices may be improved. For example, by extending the analysis to also consider existing practices, or by considering specific communication paths. In addition, further development of the underlying theory, e.g. with contextual factors and connections between distances, could also improve the specificity of the Gap Finder prescriptions, and thereby improve the effectiveness of the method. Furthermore, the Gap Finder workshop in which the assessment is presented to the project team could be improved by focusing on the main communication gaps and thereby utilise the time spent more efficiently. Finally, the method could be further strengthened by adding a step for process improvement planning, which should include additional decision makers, e.g. responsible line manager.

Future work includes adapting and applying Gap Finder to additional cases within and beyond the current scope of validity. For example, to projects that rely on document-based communication, e.g. for phase-based and distributed development projects. This will require improving and extending the measurements for artefact-related distances, and researching how tools affect requirements communication and distances. An interesting avenue to explore, is if it is possible to identify common patterns in RET profiles between related project types, e.g. projects that apply an agile development model, or distributed projects. We also plan to continue developing and refining the underlying theory including the Gap Model to cover additional case characteristics and to enable better precision in identifying improvement practices.

In conclusion, the current version of the Gap Finder can support agile software development project teams in improving their requirements communication and subsequently their requirements-test alignment in a novel way. By providing an objective view of underlying factors, i.e. distances, the method allows practitioners to take a step back and consider root causes rather than focusing on problems with existing processes. In addition, the method design and its prescriptive step are based on an empirically founded theory from a previous multi-case study. The method first assesses the actual case by measuring distances and then compares the found gaps to an existing theoretical model of best practices. In this way, the method diagnoses communication gaps of the case at hand and prescribes case-specific practices to mitigate these.

Acknowledgements We would like to thank the development team members for enabling this study by sharing their time, thoughts and office space. This work was partly funded by EASE (http://ease.cs.lth.se) and by Ericsson Research.

Appendix: The Gap Model

Table 5 shows the Gap Model and the practices that the prescriptions for the reported case are based on

Table 5 The Gap Model defined in (Bjarnason 2016), see Section 2.3 of this paper, and practices prescribed in this case study, see Section 7.2

Abstracted (APn) and detailed practices	Prescribed	Distances							
		Geo	Org	Psy	Cog	Adh	Sem	Nav	Temp
AP1 Cross-role collaboration		D	B	D	BD	D	D		D
Customer communication at all requirements levels and phases	P2								
Product manager involved in development project									
Use of a customer proxy role									
Product manager physically present to developers and testers	P1								
Development team located at customer site									
Development-near roles involved in detailing requirements									
Subsystem expert involved in requirements definition									
Collaborative definition of quality requirements	P8, P9								
Cross-role requirements review									
Small-scale development	P4								
Interaction designer in development team									
Acceptance test cases defined by customer									
Product manager reviews prototypes									
Early test involvement in development projects									
Early verification start									
Competence development	P6								
Close cooperation between Test and Development unit and roles									
Process for requirements changes involving Testing roles									
Traces/connections explicitly defined between people/roles									
Job rotation	P7								
AP2 Separate testers			I		BD	D	B		
User / Customer testing	P5								
Independent testing (relative implementation)									
Separate testing team for quality requirements									
AP3 Documentation					BD	D			
Documentation of requirement decision rationales									
Note current thinking and motivation in test cases									
Tool support for requirements and testing									
AP4 Aligning documentation structures and tracing			B	D	BD	D	D	D	
Role defined for traceability responsibility									
Feature requirements documentation									
Feature-based test plan									

Table 5 (continued)

Abstracted (APn) and detailed practices	Prescribed	Distances							
		Geo	Org	Psy	Cog	Adh	Sem	Nav	Temp
Document-level traces									
Traces between requirements and test cases									
Test cases used as requirements specification									
Same abstraction levels for requirements and test specification									
Conceptual tracing									
Tool support for tracing between requirements and test cases									
AP5 Cross-artefact reviews			B		B	D	D		
Requirements review responsibilities defined									
Test cases reviewed against requirements	P3								
Management base launch decision on test reports									
Testers re-use customer feedback									
Test-impact analysis									
AP6 Incremental software engineering					BD	D	D		D
AP7 Automated testing						D			D
AP8 Use of alignment metrics					B	D			B

References

Afzal W, Alone S, Glocksien K, & Torkar R (2016) Software test process improvement approaches: a systematic literature review and an industrial case study. J Syst Softw, 111:1–33

Angermo Ringstad M, Dingsoyr T, Brede Moe N (2011) Agile process improvement: diagnosis and planning to improve teamwork. Proc of 18th European Conf. On systems, software and service process improvement (EuroSPI'11). Communications in Computer and Information Science 172:167–178

Barmi ZA, Ebrahimi AH, Feldt R (2011) Alignment of requirements specification and testing: a systematic mapping study. Proc. 4th Int. Conf. On Softw. Testing, verification and validation workshops (ICSTW):476–485

Basili VR (1985) Quantitative evaluation of software methodology. Tech. Report TR-1519, University of Maryland, College Park, Maryland

Basili VR, Rombach HD (1988) The TAME project: towards improvement-oriented software environments. IEEE Trans Softw Eng 14(6):758–773

Benner P (1982) From novice to expert. Am J Nurs 82(3):402–407

Berntsson Svensson R, Gorschek T, Regnell B (2009) Quality requirements in practice: an interview study in requirements engineering for embedded systems. In: In international working conference on requirements engineering: Foundation for Software Quality. Springer, Berlin Heidelberg, pp 218–232

Bjarnason E (2013) Research material for gap finder evaluation study incl measurement instrument, interview guide etc. http://serg.cs.lth.se/research/experiment_packages/GapFinder/Accessed 05 May 2017

Bjarnason E, Sharp H (2015) The role of distances in requirements communication: a case study. Requir Eng 2015:1–26

Bjarnason E, Wnuk K, Regnell B (2011) Requirements are slipping through the gaps – a case study on Cause & Effects of communication gaps in large-scale software development. Proc. of 19th IEEE Int requirements engineering Conf., pp. 37–46

Bjarnason E, Runeson P, Borg M et al. (2013) Challenges and practices in aligning requirements with verification and validation: a case study of six companies. Empirical software engineering, 19(6), pp. 1809–1855

Bjarnason E, Hess A, Berntsson Svensson R, Regnell B, Doerr J. (2014) Reflecting on evidence-based timelines. IEEE Softw 31.4 (2014): 37–43
Bjarnason E, Smolander K, Engström E, Runeson P (2016) A theory of distances in software engineering. Inf Softw Technol 70:204–219
Boehm BW (1981) Software engineering economics, Upper Saddle River, –Prentice Hall
Brede Moe N, Dingsoyr T, Royrvik EA (2009) Putting agile teamwork to the test – an preliminary instrument for empirically assessing and improving agile software development. Proc of XP 2009, LNBIP 31, Springer, Berlin, Heidelberg, pp 114–123
Briand L, El Emam K, Melo WL (1995) ANSI – An Inductive Method for Software Process Improvement: Concrete Steps. Proc. of the ESI-ISCN'95: Measurement and Training Based Process Improvement, Sep. 11–12 1995, Vienna, Austria
Burstein F, Gregor S (1999) The systems development or engineering approach to research in information systems: an action research perspective. Proceedings of the 10th Australasian conference on information systems. Victoria University of Wellington, New Zealand, 1999
Chrissis MB, Konrad M, Shrum S (2007) CCMI for development, v 1.2. Guidelines for process integration and product improvement (2nd edition), SEI series in software engineering, Addison-Wesley
Collier B, DeMarco T, Fearey P (1996) A defined process for project postmortem review. IEEE Softw 13(4):65–72
Curtis B, Hefley WE, Miller S (2002) The people capability maturity model: guidelines for improving the workforce. (ISBN 0–201–60445-0). Addison Wesley Longman, Reading
Damian D, Chisan J (2006) An empirical study of the complex relationship between requirements engineering processes and other processes that Lead to payoffs in productivity, quality, and risk management. IEEE Trans Softw Eng 32(7):33–453
Damian D, Chisan J, Vaidyanathasamy L, Pal Y (2005) Requirements engineering and downstream software development: findings from a case study. Empir Softw Eng 10:255–283
Derby E, Larsen D (2006) Agile retrospectives: making good teams great! Pragmatic Bookshelf, 2006
Drury M, Conboy K, Power K (2011) Decision making in agile development: a focus group study of decisions and obstacles. Proc. of Agile Conference 2011:39–47. https://doi.org/10.1109/AGILE.2011.27
Dybå T (2000) An instrument for measuring the key factors of success in software process improvement. Empir Softw Eng 5:357–390
Dybå T (2005) An empirical investigation of the key factors for success in software process improvement. IEEE Trans Softw Eng 31(5):410–424
Ferguson RW, Lami G (2006) An empirical study on the relationship between defective requirements and test failures. Proc of 30th annual IEEE/NASA software engineering workshop SEW-30 (SEW'06)
George M (2002) Lean six sigma: combining six sigma quality with lean production speed. McGraw-Hill, New York
Gotel O, Finkelstein A (1994) An analysis of the requirements traceability problem. Proc. First Int Conf. Requirements Eng., pp. 94-101
Harter DE, Kemerer CF, Slaughter SA (2012) Does software process improvement reduce the severity of defects? A longitudinal field study. IEEE Trans Softw Eng 38(4):810,827, July-Aug. 2012. https://doi.org/10.1109/TSE.2011.63
Hevner R, March S, Park J, Ram S (2004) Design science in information systems research. MIS quarterly 28.1: 75–105. https://www.in.thnuernberg.de/professors/holl/personal/hevner_designscience_isres.pdf
Humphrey WS (1989) Managing the software process. Addison-Wesley, SEI Series in Software Engineering
Humphrey W (1997) Managing technical people: innovation, Teamwork, and the Software Process, Addison-Wesley
ISO/IEC (2004–2011) ISO/IEC 15504 Information Technology – Process Assessment, parts 1–10
Kandt RK (2009) Experiences in improving flight software development processes. IEEE Softw 26(3):58–64
Kukkanen J, Vakevainen K, Kauppinen M, Uusitalo E (2009) Applying a systematic approach to link requirements and testing: a case study, proc of asia-pacific software engineering conference (APSEC '09):482–488
Lavallée M, Robillard PN (2012) The impacts of software process improvement on developers: a systematic review. Proc of 34th Int. Conf. On software engineering (ICSE), pp.113–122. https://doi.org/10.1109/ICSE.2012.6227201
Martin R, Melnik G (2008) Tests and requirements, requirements and tests a Möbius strip. IEEE Softw 25(1):54–59
Post H, Sinz C, Merz F, Gorges T, Kropf T (2009) Linking functional requirements and software verification. Proceedings of 17th IEEE international requirements engineering conference, pp. 295-302
Ramesh B, Cao L (September 2010) Baskerville R (2010) agile requirements engineering practices and challenges: an empirical study. Inf Syst J 20(5):449–480
Robinson H, Segal J, Sharp H (2007) Ethnographically-informed empirical studies of software practice. Inf Softw Technol 49:540–551
Robson C (2002) Real world research. 2nd ed. Blackwell Publishing, Hoboken
Rogers Y, Sharp H, Preece J (2011) Interaction design: beyond human - computer interaction, 3rd Edition. Wiley, Hoboken
Runeson P, Höst M, Rainer A, Regnell B (2012) Case study research in software engineering – guidelines and examples. Hoboken, Wiley

Sabaliauskaite G, Loconsole A, Engström E, Unterkalmsteiner M, Regnell B, Runeson P, Gorschek T, Feldt R (2010) Challenges in Aligning Requirements Engineering and Verification in a Large-Scale Industrial Context Proceedings of REFSQ 2010

Unterkalmsteiner M, Feldt R, Gorschek T (2014) A taxonomy for requirements engineering and software test alignment. Accepted for publication in ACM Transactions on Software Engineering and Methodology

Uusitalo EJ, Komssi M, Kauppinen M et al. (2008) Linking requirements and testing in practice. 16th IEEE Int Requirements engineering Conf, NJ, USA, pp. 265–270

Dr. Elizabeth Bjarnason is a senior lecturer with the Software Engineering Research Group, Lund University, Sweden. Her main research interests lie within empirical software engineering, in particular requirements communication, requirements-test alignment and software process improvement. Bjarnason combines a long experience of software engineering in industry with empirical research in the field in close collaboration with industry partners. Contact her at elizabeth@cs.lth.se .

Helen Sharp is Professor of Software Engineering at the Open University, UK. Her research investigates professional software practice with a focus on human and social aspects of software development and she has been studying agile practice since 2000. Sharp has led multi-disciplinary research projects into software practice

with partners in the UK and abroad, and conducts her research exclusively in-situ with software practitioners in their industrial context. Sharp is joint author of one of the leading HCI textbooks, *Interaction Design* now in its 5th edition. She is also on the editorial board for *EMSE* and *JSS*, serves on the Advisory Board for *IEEE Software* and will chair ICSE's software practice track in 2019.

Björn Regnell is Professor in Software Engineering at the Faculty of Engineering, LTH, Lund University, Sweden. He has contributed to several software engineering research areas including requirements engineering, software quality, software product management and empirical research methods in software engineering. Regnell has been Steering Committee Chair and Program Chair of International Working Conference on Requirements Engineering: Foundation for Software Quality (REFSQ), Program Chair of International Conference on Software Business (ICSOB), and a member of the Program Board/Committee of International Requirements Engineering Conference (RE). He is a member of the Editorial board of the Requirements Engineering Journal. Prof. Regnell has published more than 80 peer-reviewed articles in journals and conferences. He has edited several special issues in journals and proceedings and he is co-author of several books including the widely cited "Introduction to Experimentation in Software Engineering" and "Case Study Research in Software Engineering - Guidelines and Examples".

Affiliations

Elizabeth Bjarnason[1] · Helen Sharp[2] · Björn Regnell[1]

Helen Sharp
helen.sharp@open.ac.uk

Björn Regnell
Bjorn.regnell@cs.lth.se

[1] Department of Computer Science, Lund University, Lund, Sweden

[2] Computing and Communications Department, The Open University, Milton Keynes, UK